Dejan Ristanović

računari u vašoj kući

Specijalno izdanje časopisa „Galaksija"
Izdaje BIGZ — DUGA
Januar 1984. Cena 200 D



izbor i primena računara
pregied programa, časopisa i knjiga
kompletno uputstvo
za samogradnju kućnog kompjutera
umetak na 32 strane: programiranje za početnike

računari u vašoj kući

Autor: Dejan Ristanović

Blok o računaru „galaksija“: Voja Antonić

Specijalno izdanje časopisa „Galaksija“

Januar 1984. Cena 200 D

Zahvaljujemo Anđelku Zgorelcu, London, na pomoći prilikom realizacije ovog specijalnog izdanja.

sadržaj

Izdaje
Beogradski izdavačko-grafički zavod
OOUR Novinska delatnost „Duga“
11000 Beograd
Bulevar vojvode Mišića 17

Telefoni
650-161 (redakcija)
650-526 (prodaja)
651-793 (propaganda)

Generalni direktor
Gojko Zečar

Direktor OOUR „Duga“
Zoran Milošević

Glavni i odgovorni urednik
Gavrilo Vučković

Urednik izdanja
Jova Regasek

Likovna i grafička oprema
Dušan Mijatović

Redakcija časopisa „Galaksija“
Tanasije Gavranović, pomoćnik glavnog i odgovornog urednika
Esad Jakupović, zamenik glavnog i odgovornog urednika
Aleksandar Milinković, urednik
Jova Regasek, urednik
Zorka Simović, sekretar redakcije
Gavrilo Vučković, glavni i odgovorni urednik

Izdavački savet „Galaksije“
Prof. dr Brana Dimitrijević, Ivan Gajić, dr Branko Jovičić, prof. dr Branko Lalović, Vigor Majić, prof. dr Dejan Milovanović, Milan Milutinović (predsednik), Đokica Petković, Rade Petrović, Dušan Radović, Jova Ragasek, dr inž. Zlatko Rendulić, dr Petar Strugar, Dimitrije Tasić (zamenik predsednika), Gavrilo Vučković

Štampa
Beogradski izdavačko-grafički zavod
11000 Beograd, Bulevar vojvode Mišića 17

Žiro-račun kod SDK
60802-603-17132

Devizni račun kod Beogradske banke
60811-620-16-82701-999-01066

Za inostranstvo cena dvostruka
(400 D, 3 US $, 9 DM, 60 Sch, 7 Sfrs, 28 Ffrs, 2 Lstg)

Na osnovu mišljenja Republičkog sekretarijata za kulturu broj 413-77/72-03 i „Službenog glasnika“ broj 26/72, ovo izdanje oslobođeno je poreza na promet

Snimak na naslovnoj strani
Ivan Ivanov

Kućni računari stiču sva više poklonika u čitavom svetu. Smatra se da je do sada prodato nekoliko miliona kompjutera namenjenih amaterima — ljudima koji računare koriste za igru, različite poslove u kući, u nauci, obrazovanju i manjim firmama. Već sada je sasvim izvesno da će kućni računar veoma brzo izrasti na samo u najmasovniji hobi nego i u sredstvo bez koga neće moći ni da se zamisli svakodnevni život. Računar je univerzalna alatka, kao poluga i kao točak, i njegova primena ograničena je jedino maštom i znanjem njegovog vlasnika.
U našoj zemlji kućni računari nemaju, doduše, tako impresivno uporište kao u svetu, ali broj ljubitelja računarske tehnike i vlasnika kućnih kompjutera ni kod nas nije za potcenjivanje. On bi, svakako, bio znatno veći da su okolnosti oko popularizacije i, naročito, oko nabavke kućnih računara barem malo povoljnije. Veoma je oskudna literatura na srpskohrvatskom jeziku koja govori o prirodi kućnih računara i koja sadašnjim i budućim vlasnicima može da se nađe pri ruci kao pouzdan oslonac, savetnik i priručnik u njegovoj avanturi izbora, nabavke, upotrebe, primene i programiranja računara. Osnovna namera našeg izdanja jeste da popuni upravo tu prazninu. Na žalost, samo u informativnom smislu. Plašimo se da ono neće ni za dlaku promeniti tretman računara kod naših carinika.
Svoju dvogodišnju aktivnost na popularizaciji kućnih računara u „Galaksiji" u ovom izdanju smo krunisali i jednom sasvim konkretnom akcijom — projektom za samogradnju originalnog kućnog računara. Ova akcija ima, pre svega, za cilj da omogući svim ljubiteljima računara spretnih ruku da najzad dođu do svoja prve mašine, ali i da razbije mit o računaru kao složenom tehničkom proizvodu. Računar „galaksija" je projektovao jedan jedini čovek, Voja Antonić, za nešto više od mesec dana, a od prve ideje do pojava prvih primeraka na tržištu prošlo je jedva nešto više od šest meseci. Naš računar, bez ikakvih koncepcijskih ili tehničkih izmena, ovih dana ulazi i u serijsku proizvodnju, a njegovi proizvođači, „Elektronika Inženjering" i „Zavod za učila i nastavna sredstva", zatrpani su porudžbinama na samo pojedinaca nego i škola.
Ovo izdanje je koncipirano tako da predstavlja zaokruženu celinu. Ono se, pri tom, na bavi sociološkim, kulturološkim ili psihološkim aspektima računarske tehnike. Njegova osnovna namera je da prikaže računar u akciji, na njegovom svakodnevnom poslu, sa svim bogatstvom primena i svim — ograničenjima. To je, verujemo, najbolji način da se računari upoznaju i — zavole. Neko će možda primetiti da računare i ne treba voleti previše. To je verovatno tačno. Ali se na njih treba naviknuti, jer će život bez njih u kompjuterizovanom svetu neposredne budućnosti biti sasvim nemoguć.

Redakcija

1 računari na sto načina

upotreba računara

Postoji ipak jedna suštinska razlika između kompjutera i bilo kog drugog uređaja u domaćinstvu: svaki uređej se kupuje da bi obavljao neku jasnu i sasvim određenu funkciju (nikome, na primar, neće pasti na pamet da koristi frižider da bi gledao TV program). Kada nabavljate računar, kupujete mašinu koja baš ništa ne zna! Kada, međutim, u njega upišete odgovarajući program, on počinja da podučava decu, igra šah, proračunava troškove, balansira čekovnu knjižicu... Sve to u jednoj stvari? Da, i još mnogo čega drugog: računar može da uradi sva što poželite samo ako umate da mu objasnite keko da to uradi.

Takvo objašnjenja se zova program. Program je skup naredbi (u stilu: „prikaži na ekranu") pisan na nekom od jezika koji kompjuten razumaju. Dok „veliki" kompjuterski sistemi koriste moćne i složene jezike koji nose imena fortran, kobol, paskal, ada i slično, svi stoni računari sa sporazumevaju sa svojim vlasnicima na jaziku koji nosi ime bejzik. Iako je reč bejzik, u stvari, skraćenica, ona odmah asocira na osnovnu namenu ovog jezika: on predstavlja osnovu i početak za sve nove ljubitelje računara i omogućava im da naprave prve korake u programiranju i docnije lakše savladavaju složenije kompjutere.

Ako nabavite računer, svakako da u prvom trenutku nećeta biti u stenju da pišete programa za njaga. To, ipak, na znači da će računar biti neiskorišćeni deo kućnog inventara. Na tržištu (pa čak i na jugoslovenskom tržištu) se nalazi mnoštvo komercijalnih programa koji su spremni da budu upisani u vaš kompjuter. Takav program ćete dobiti na običnoj magnatofonskoj kaseti. Ako je stavite u kasatofon i počnete da je slušate, čućete veoma naprijatan zvuk koji je, na prvi pogled, samo ravnomerno zujanja. Taj zvuk je vešam računaru sasvim familijaran. dovoljno je da pomoću posabnog kabla považete računar sa kasetofonom i otkucate jednu reč pa će kompjuter početi strpljivo preslušavanje keseta. Posle nekoliko minuta, na ekranu će se ponovo



Od najranijih nogu. Računarima mogu do rukuju čak i deca predškolskog uzrasta

pojaviti neka poruka koja označava da je takst pročitan i shvaćen. Zatim ćete otkucati još jednu reč i računar će početi da radi ono što je bilo upisano na kaseti. Tada će započeti partija šaha, neka nova video-igra, računar će vam postevlti neki matematički problem ili će, jednostevno, na akranu započeti neka vrata vrste ctranog filma.

Upotreba komercijalnog programa je krajnja jednostavna. Najčešće možete da se služite palicom kao kod popularnih partija tenisa uz korišćenja „video-igara". Pokatkad se ove palice zamenjuju pritiscima na tastere (značete, na primer, da taster A označava kretenje levo, taster L kretanja desno, taster Z gora, B dola, a da se razmaknicom bacaju bombe). U drugim slučajevima „razgoveraćete" sa računarom tako što ćete odgovarati na pitenja koja postavlje kucajući standardna angleska ili srpskohrvatska rečenica, ili birati jedan od naslova u katalogu koji je ispisan na akranu. Za korišćenja ovakvih programa nije, dekla, potrebno nikakvo posebno znanje — dovoljno ja da pažljivo pročitete priloženo uputstvo i eksperimentišeta — probajući sve alternative ubrzo ćete upoznati sve tajna programa i početi ozbiljno da ga upotrebljavate.

Iako komercijalnim programima možete da se zabavljate duže vreme, pre ili posle ćete uvideti da ovo zebava nija

Do pra aamo desetak godine rukovanje računarima je bilo dostupno semo vrlo uskom krugu visoko obrezovenih stručnjeke: progremere i oparatore. Računar je več tada bio moćna mešine, ali njegove cene nija dopuštele prosečnom čoveku de pomisli ne nebevku ovekve neprave. Rezvoj elektronike i minijaturizecije komponeneta su učinili svoja: dimenzije i cene računara su toliko amenjene de su kompjuteru širom otvorene vreta u svakodnevni život.

kreativna, pa četa požalati da saznete kako su programi napravljeni i da napišete neki samostalni. Tada će vas zainteresovati razna škole bejzika, priručnici, tekstovi u časopisime . .

Da bista naučili bejzik nija vam potrebno nikakvo posebno matamatičko-tahničko predznanja. Proces učanja je naobično zanimljiv, eli bismo vas prevarili kada bismo rakli da je lek. Stara narodna poslovica kaže da „bez muka nema neuke'': svaku komendu čate sevledevati tek posle odredanog proučavanja priručnike i isprobavanja njanog dajstva na računaru. Prvi programi će biti vrlo jednostavni, ali će se i u njima provlačiti greška koje čete morati samostalno da pronalezite. Kompjuter, naime, uvek doslovno izvršava vaše naredbe, jer nije u stanju da „razmisli'' o toma da li su one svrsishodna. Ako mu, na primer, naložite da stalno izvršava istu naredbu, on će je strpljivo izvršavati sve dok ga ne prakinete ili dok vaša grupa na dođe na red za isključenje struje. Zato ja jedan frustrirani početnik izrekeo lapu misao: „programiranja je teško jadino zato što kompjuter nikada na radi ono što ja hoću nego ono što mu kažem!''

Smatramo de ja za savladavenje tehnike programiranje neophodno samo mnogo dobre volje. Proces učenja se kod raznih ljudi završava na razne načine: neki su zadovoljni kada uspeju da sesteve program pomoču koge ča njihov računar sastavljati optimalne sistama sportska prognoza, drugi prestaju da reda kada nauče dovoljno da sastave progrem koji će proračunati komponente nekog alaktronskog sklopa, treći kada sastava programe koji će pomoči najmladim članovime porodice de zavola matamatiku i geografiju . . . Naki se ne tom putu na zaustevljeju: nestevljeju da rada sve dok ne ovladaju čak i najskrivenijim tajnama njihovog kompjutare, a zatim nabavljaju novi da bi ga podvrgli istom postupku. Tako nastaju kompjutarski stručnjaci. Programiranje ja, kada sa njime ozbiljno i profasionalno bavimo, neuka koja zahteva nešto talenta i mnogo rada. Školovani programari su osposobljani za sastavljanje izuzetno složanih programa kekvi sa, na primer, neprekidno nalaze u mamoriji vašeg računara omogućavajuči mu normelen rad.

Kako da razlikujemo „zagriženog ljubitelja računara'' (u stranoj literaturi, a u poslednja vrema i kod nes, ovakva ličnost se naziva „hackar'', što bi moglo da sa praveda kao „kljucalo''; izraz ni u kom smislu nije pogrden) od „normalnog'' korisnika? Jadnostavno — za „hekare'' ja pisanje progreme semo sabi cilj, pa program postaja potpuno nazanimljiv kade se isprave sve greške i tako doveda u konačnu formu. Nasuprot toma, za „normalnog'' korisnika računare pisanje programa postaje svrsishodno tek eko tej program može da se primeni u njagovom životu i tako mu olekše neki od svakodnavnih poslova, keo što ja pisanje pisama, sređivanje biblioteke knjiga, vođenje rokovnika . .

Namamo nikekvo pravo da kažamo da je jeden od ova dva pristupe računarima „pravilan'' ili „pogrešen'', tim pre što ni „hakar'' ne ostaje uvek „heker''. Početno oduševljenje koje nas obuzma kada nebevimo prvi računar svakako če učiniti da u njagov ekren gladamo naprekidno po dvadasatak časove dnevno u toku nakoliko sedmica. Sledi pisanja programa koje ime za cilj proveravanje naših novostečenih znanja. Sledeča je „hakerske feza'' u kojoj pišemo programe zato što nam je to zedovoljstvo i zabava. Te programe nejčešće nudimo prijeteljima ili kompjuterskim klubovima sa kojima saradujemo (oni u kojima kompjuterska groznica na može da suzbije poslovni duh mogu dobro da unovče programa, naročito eko su inventivni, originelni i dobro sastavljani) ili ih, jednostevno, odlažemo na magnetofonska kesete „ze buduću upotrabu''.

Ako ikada izađemo iz ove faze, računar ča za nas postati vrlo korisna naprava koja moža da sa upotrebi za milion stvari u kući i na poslu, koja može da predstavlja izvanrednog partnara u igrama i tako razbija dosadu i koju, nejzad, možamo da „učimo nečem novom'' svaki put kada imamo volje za pisanja programa. Teda će nam najveće zedovoljstvo biti upoznavanja novih modala računera i poredenje njihovih mogučnosti sa mogučnostima drugih modela. Kako vrame ida dalje, upoznavanja novih kompjutera če (ovde bismo, verovali ili ne, redo dodeli jedno „na želost'') biti sve kratkotrajnija zebeve, eli ćemo zato početi da razmišljemo o tome keko bi trebao de izglede „ideelni'' računar i „idealni'' programski jezik . .

Šta dolazi posla toga? Autor ovog izdanja ne bi želao da mu odgovorite ne ovo pitanja: pisanje ovih redova je bilo divna prilika de ponovo proživi posladnjih pet godine, ali bi vaš odgovor svakako učinio da sledečih pedeset izgube draž naizvesnog.

Od nas se, ipak, očekuja da vam kažemo u kavom čate se društvu neči eko postanate „ljubitelji računara''. Koliko ja, pre svega, to društvo brojno? Smatra se de u svetu tranutno ima nekoliko milione stonih računara i da oko dve stotine hiljade ljudi ima dovoljno znanje da piše jednostavnije programa za njih, dok se nakih hiljadu ubraja u „elitne programere''. Keko stoji stvar u Jugoslaviji? „Galeksije'' se iskreno neda da ljubitelje računere, osim vas, ima bar još 29999 (ovo izdenje se štampa u 30000 primereka) i da če ih u budućnosti biti još mnogo više. Ljubitalji računere se okupljaju u okviru nekoliko klubova programera, kupuju računare i programe za njih preko melih oglasa, udružuju se redi nabavke ne baš jeftine strana literature . . . U čitav tej posao je ugrađeno mnogo više antuzijazma nago organizovane društvene akcije.

Niko ne spori da su računari značajni u savramenom svetu i de je neophodno obrezovati kvalitatnog kompjuterskog stručnjaka. Pri tome se, na žalost, uloga stonog računare u mnogo čemu umanjuje; rezloge trebe tražiti u činjenici de mnogi naši autoriteti (ma koliko stručni bili) ne shvataju da stoni računar više nije nemoćna sprava već univerzalna alatka ogromnih mogućnosti koji je pristupačan mnogo čijam džepu. Keda su več naši računarski centri prezagušeni poslom toliko da u njima nama mesta ni za koga osim za zaposlana i kade se na fakultetima koji obrazuje programera mari svaki sekund koji student provede ze terminalom i broji svaki list pepira koji štampa, trebalo bi liberalnijim carinskim propisima i stimulisenjem domečih proizvodače pomoči svekome ko žali da upozna računere o svom trošku.

2 jabuka iskušenja izbor računara

Najvažniji paramatar pri izboru računara su, svakako, vaše potrebe. Njih treba proceniti na osnovu trenutne situacije ali i bliska budućnosti: ne traba da kupujate računar koji jedva zadovoljava vaše sadašnja potrebe (jer će u bliskoj budućnosti potraba svakako porasti), ali ni da plaćata sistem ogromnih mogućnosti koje nećete imati gda da primenjujeta.

Džepni ili stoni?

Stoni računar, u principu, pruža veće mogućnosti za manja novca. Ima, međutim, dosta korisnike kojima je računar potreban na terenu (pomanimo samo arhitakte i građevinara), u školi ili na fakultetu. U većini takvih slučajeva dovoljno je nabaviti neki džepni računar sa obiljem matematičkih funkcija koji nema mogućnost programiranja (takvi su danas prilično jaftini) i, za kućnu primenu, dobar stoni računar. U nekim slučajevima, međutim, potreban je prenosiv računar čija memorija može da primi nekoliko programa koji se koriste „svuda i na svakom mestu". U tom slučaju se opredelite za džepni računar i pokušajte da sastavite ili napravite neophodne programa.

Globalno posmatrano, računari se koriste u naučno-tahničkim proračunima, svakodnevnim poslovima, za vežbanja programiranja i za igranje (postoje, jasno, i stotine drugih primena). Džepni računari mogu vrlo dobro da posluže za naučno-tahničke proračune (iako je manja brzina njihovog rada unekoliko prepreke) i za vežbanja u programiranju. Poslovna primena im nije jaka strana, a što se igara tiće — sve zavisi od veštine programera. Džepni računari, naime, imaju najčešće ekran koji sadrži samo jednu liniju, pa su akcione igre pomalo monotone. U logičkim igrama džepni i stoni računari su praktično ravnopravni.

Pri donošenju odluke o kupovini računara treba se čuvati jedne velike greške: ne kupujte loš džepni i loš stoni računar. Ako su materijalna sredstva ograničena (a najčešće jesu), kupite jedan dobar kompjutar. Jer, računar je ipak računar, a razlike između džepnih i stonih, za veštog programera, nisu velike.

Računari i klase

Cena je, jasno, jedan od osnovnih paramatara koji traba razmotriti. Što je računar bolji, to je skuplji, ali to ne znači da ćemo, ako imamo novca, kupiti najskuplji postojeći modeli. U cenu ne ulazi samo procena mogućnosti računara, značajan paramatar



je ime firme koja ga proizvodi (renomirane firma kao što su IBM i Hewlatt-Packard daju sebi pravo da svoje proizvode prodaju dvostruko skuplja od ostalih kompanija), država u kojoj se računar kupuje, broj prodatih primaraka . Da bi se unekoliko klasifikovala mogućnosti (i cene) računara, često se uvodi podela na ekonomsku, srednju i višu klasu. Ni jedna od ovih klasa nije strogo odvojana od drugih i postoje modeli koji su na prelazu iz ekonomske u srednju (nema ih mnogo) i iz srednja u višu klasu (takvih je dosta).

Računari iz ekonomske klase se prepoznaju po ceni manjoj od 150 funti (obzirom da je pri pripremanju ovog poglavlje korišćena uglavnom engleska literatura, cena su uglavnom u funtama). Ovi računari su dobri za one koji žela da nauče programiranja, sastave prve programa i zabavljaju se komercijalnim programima za igre. Primena računara iz ekonomska klasa na naučno-tahničke proračune je moguća, ali ne bez izvesnih taškoća: loša tastatura onemogućava brzo unošenja eksperimentalnih podataka, računar često prilično sporo radi pa proračun može da potraje, za priključivanja štampača i periferijskih uređaja potrebni su ne baš jeftini intarfejsi. . . Kompjutari iz ove klasa nisu pogodni za poslovne primena u kojima se od računara zahteva obrada velika količine informacija.

Srednja klasa je vrlo širok pojam i obuhvata računara koji koštaju 200-700 funti. Ovi računari su veoma pogodni praktično za sve primane, pogotovu zato što mogu lako da se proširuju dodavanjam neke parifarijske opreme. Njihove tastature su profesionalna, mamorija prihvatljivo velike, poseduju par programskih jezika i pristojnu programsku podršku. Za ona koji nameravaju samo da se igraju i uče programiranja, kupovina računara iz srednja klase predstavlja neku vrstu luksuza, za one koji namaravaju da koriste računar u neke „ozbiljne" svrhe (pogotovu ako će računar obavljati posao koji donosi novac) srednja klasa je idealno rešenja.

Računari iz više klase se praktično ne razlikuju od računara iz srednja. No, oni su opremljani nekim perifarijskim uređajima kao što su disk-jedinice i sopstveni monitor. To znači da ovi računari predstavljaju zaokružen kompjuterski sistem izuzetno pogodan za poslovne primana (na primer, vođenja poslovanja nekog objekta male privreda). Činjanica je da računari iz srednja klase, kupovinom dodatnih uređaja, mogu da pređu u višu, odakle sledi da je nešto povoljnije kupiti računar iz srednje klase, upoznati njegove mogućnosti i proširiti ga ukoliko za tim postoji potreba. Od ovog zaključka se odstupa samo u jednom slučaju: ukoliko planiramo upotrebu računara za neke poslove i sigurni smo da će nam disk-jedinice, štampač i velika memorija biti potrabna, bolje je da ih odmah nabavimo: ne samo što će sistem biti lakši za manipulisanja (sva se komponente nalaze u okviru samog računara) nego ćemo i jeftinija proći. Ukoliko želimo da koristimo računar za igranja, nije dobro da kupimo model iz više klase: za njega su uglavnom pisani „ozbiljni" programi, pa na tržištu nećemo moći da nađemo igre koje nas interesuju.

Kada procenimo potreba i odlučimo da kupimo stoni računar iz neke od ove tri klasa, pristupamo pregledu tabela. Najpre ćemo odbaciti računare koji nam ne odgovaraju (skuplji su ili jeftiniji nego što želimo), a zatim ćemo uporedivati ostala karakteristike. Pri tom ćemo naići na određen broj reči koje nam se čine besmislenim (ukoliko ste početnik u svetu računara, verovatno su vas i reči: tastatura, memorija, disk-jedinice itd. koje ste pročitali u dosadašnjem takstu, pomalo zbunila). Pošto je izbor računara veoma ozbiljan posao, ne smemo da dopustimo da nam neki od ovih pojmova ostanu nejasni: možda baš oni predstavljaju presudni paramatar na osnovu koga ćemo moći da se opredelimo između dva modela.

Memorija

Jedan od osnovnih paramatara pri procenjivanju računara je njegova memorija. Računar memoriju koristi za smeštanja svih podataka koje mu dajete i za programe koji omogućavaju obradu tih podataka. U našoj tabeli ćete naći dve podatka: ROM računara i RAM.

ROM (skraćenica od Read Only Memory ili, u prevodu, memorija koja može samo da

Ljubitelji računara u Jugoslaviji nisu baš u najprijatnijem položaju. Uvoz računara, kao i svega drugog skupljeg od 5000 dinara, nije dozvoljen. Tako je većina onih koji žele da nabave kompjuter upućena na neko izigravanje zakona: ubeđivanje nekog rodaka na privremenom radu u inostranstvu da sledeće godine ne donosi televizor, video-rekorder ili kino projektor nego računar, pronalaženje nekog povratnika iz inostranstva koji ima pravo na uvoz ili, jednostavno, šverc. Takav potencijalni vlasnik računara mora unapred da odluči koji će računar da kupi pošto nema mogućnosti da vidi sve „kandidate" na delu. Koliko nam je poznato, ni jedna naša radna organizacija se ne bavi uvozom računara, strani predstavnici donose malo ili nimalo opreme na sajamske štandove, literaturu nije lako nabaviti i za nju je potrebno odvojiti prilično deviza . . . Odluka o konačnom izboru može da bude donesena samo posle čitanja uporednih prikaza različitih modela i zrelog razmišljanja.



se čita) služi za smeštanje programa koje dobijate uz računar. Ti programi se koriste za normalan rad računara. Ukoliko vaš budući računar, na primer, „zna" konstantu PI, ona je ubeležena negde u njegovom ROM-u. ROM se, kao i RAM, meri kilobajtima. Kilobajt je jedinica za količinu informacija i označava koliko znakova može da stane u memoriju računara. U jedan kilobajt memorije staje 1024 slovna znaka. Na stranu kucanu sa proredom stane oko dva kilobajta informacija.

U ROM ne možete da upisujete vaše programe, ali on pokazuje koliko različitih naredbi vaš računar poznaje. Računar koji ima ROM od 8 kilobajta ima daleko manje mogućnosti od onoga koji ima ROM od 16 kilobajta. Većina današnjih računara ima ROM od 12-18 kilobajta, koji je, uglavnom, dovoljan za realizovanje vrlo složenih programa. Ipak, u tabeli ćete naći i nekoliko računara koji imaju ROM od svega dva kilobajta. Ti računari u ROM-u drže samo najosnovnije programe, a pre početka rada sa kasetofona moraju da se učitaju ostali. Sa stanovišta iskusnog korisnika računara kompjuter sa malim ROM-om je zgodan obzirom da on može po potrebi da menja programski jezik na kome radi. Međutim, za početnika je nepoželjno da svaki put kada uključi računar mora da učitava program koji omogućava računaru da obavlja osnovne računske radnje. Ne bismo vam savetovali da kupujete računare koji imaju ROM manji od 10 kilobajta.

Neki računari dopuštaju proširivanje ROM-a. U računar se priključuju posebni moduli kojima se ROM proširuje i tako se računaru dodaju nove naredbe i povećavaju njegove mogućnosti. Računar Atom Acorn, na primer, može da radi samo sa celim brojevima. Priključivanjem dodatnog ROM-a od 4 kilobajta ovaj računar dobija naredbe za rad sa razlomcima i brojevima u „pokretnom zarezu" (brojevi koji nemaju fiksiran broj decimalnih mesta). Svi podaci u našoj tabeli dati su za računare bez ikakvog dodatnog ROM-a, ali je mogućnost proširenja spomenuta. U okviru procene vaših potreba treba da se odlučite i za kupovinu potrebnih dodataka među kojima RAM i ROM moduli dolaze na prvo mesto. Ukoliko želite da pišete matematičke programe, računar koji radi samo sa celim brojevima neće moći mnogo da vam pomogne, bolje je da odmah nabavite dodatni ROM. Važno je informisati se i o načinu ugradnje dodatne memorije: kod nekih računara dodatni ROM se jednostavno priključuje u neku utičnicu (tzv. „port"), a druge je potrebno rasklapati da bi se čip zalemio na pravo mesto. Ukoliko nemate praktičnog iskustva u elektronici, nije poželjno da se odlučite za drugu alternativu — možda ćete za vreme ove „operacije" oštetiti računar.

Neki računari imaju „ogroman" ROM od 32 ili 48 kilobajta. Važno je saznati zašto je ROM toliko veliki: u njemu su sigurno smešteni neki specijalizovani programi koji za vas mogu da budu korisni ili sasvim nepotrebni. Ukoliko, na primer, ne planirate nabavku štampača, ne kupujte računar sa velikim ROM-om u koji je ugrađen „Text-editor" (program za obradu i štampanje teksta) — tako plaćate nešto što je za vas sasvim beskorisno.

RAM (skraćenica od Random Access Memory, odnosno memorija kojoj se može slobodno pristupati) nesrećno je izabran termin koji označava memoriju koju korisnik može slobodno da popunjava. U ovu memoriju se smeštaju programi koje korisnik otkuca, podaci koje ti programi obrađuju i informacija neophodna za pravilan rad računara. I RAM se meri kilobajtima — prvi stoni računari su imali RAM od 2 — 16 kilobajta, a danas je 16 kilobajta minimalna konfiguracija. Neki računari imaju RAM koji se meri stotinama kilobajta, pa čak i megabajtovima (jedan megabajt ima $2^{10} = 1024$ kilobajta). Ne treba unapred otpisati računare koji imaju mali RAM — on najčešće može da se proširi, a može čak i da bude dovoljan za vaše potrebe. Evo nekoliko podataka o tome šta sve može da uradi računar sa 16 kilobajta memorije:

- Da „zapamti" dve gusto pisane strane u „Galaksiji" (pod pretpostavkom da na tim stranama nema slika) ili 5-6 standardnih strana (kada se izostave slike, podnaslovi i slično).
- Da „reši" sistem od 85 jednačina sa 85 nepoznatih.
- Da nađe matricu inverznu matrici 60·60.
- Da razvrsta po azbučnom redu 1500 reči, od kojih svaka ima po 10 slova.
- Da igra šah, bakgemon, othello'/reversi, složi mađarsku kocku i pobedi neiskusnog korisnika u praktično svim logičkim igrama.

Za mnoge primene 16 kilobajta je sasvim dovoljno — verovatno nećete nikada sastaviti program koji bi zauzimao više od 10 kilobajta memorije, a ukoliko ga i budete sastavili, nećete se odlučiti da ga otkucate (za kucanje takvog programa biće vam potreban bar jedan ceo dan). Ipak, mnogi korisnici se odlučuju za kupovinu gotovih programa koji omogućavaju rad sa datotekama, komponovanje teksta i slično. Ovakvi programi uzimaju veliki deo RAM-a, tako da ono što ostane najčešće nije dovoljno za primenu tako složenog programa. Takvi korisnici će morati da proširuju memoriju svog računara pomoću RAM modula. RAM moduli se ponekad priključuju na port računara, ali se najčešće ugrađuju u njega. Prilikom kupovine računara dodatni RAM modul će, verovatno, ugraditi prodavac ili ćete, ukoliko modul docnije kupite, morati sami da ga ugradite. Ovog ugrađivanja se ne treba mnogo bojati (čip se jednostavno postavi u podnožje), ali ga ipak treba prepustiti stručnom licu ako je to ikako moguće.

Realno je proširiti memoriju računara do 32 kilobajta. Za neke primene (posebno obradu teksta) nju treba proširiti i do 64 kilobajta, ali vam ne bismo predložili da to odmah radite. Najpre nabavite računar i upoznajte ga. Tada ćete steći osećaj o tome šta su 32 kilobajta i odlučiti da li vam zaista treba više. Ukoliko na pitanje odgovorite potvrdno, lako možete da naručite memorijski modul. Ukoliko uvidite da vam više memorije ne treba, uštedećete dosta novca — memorijski čipovi su vrlo skup deo računara.

Jezici

Gotovo svi kućni računari rade na bejziku — kompjuterskom jeziku opšte namene. Ukoliko je u našoj tabeli napomenuto da računar radi na bejziku, za upotrebu ovog jezika nije potrebno kupovati nikakve dodatke — u samom ROM-u računara je ugrađen program koji prevodi rečenice koje korisnik otkuca (tzv. bejzik naredbe) na oblik koji računar može da shvati i koristi.

Neki računari, pored bejzika, imaju i druge programske jezike, najčešće paskal. Paskal je programski jezik namenjen mate-

matičarima i inžinjerima i u literaturi ćete naći podatak da se paskal **kompajlira** (prevodi) a bejzik **interpretira.** Šta to znači? Ako je u računar unet program u paskalu, on će, pre njegovog izvršavanja, morati da čitav taj program, rečenicu po rečenicu, prevede u oblik koji je računaru pristupačan. Posle toga, korisnik može da zahteva da se prevedeni program izvrši. Kod bejzika (i drugih jezika koji se interpretiraju), situacija je nešto drukčija: računar nailazi na neku bejzik naredbu, prevodi je i izvršava, a zatim „zaboravlja" prevod te naredbe i na isti način obrađuje sledeću. Sa aspekta korisnika i jedan i drugi tip jezika imaju prednosti i nedostatke. Jezici koji se kompajliraju bolje koriste mogućnosti računara jer programi rade mnogo brže, ali je zato posle svake sitne ispravke neophodno prevoditi program. Bejzik programi rade daleko sporije, ali je ispravljanje i prilagođavanje programa brzo i jednostavno. Sve u svemu, za onoga ko se prvi put susreće sa računarima i za iskusnog korisnika koji nema potrebu za velikim računanjima, bejzik računari predstavljaju racionalnije rešenje. Računari koji koriste paskal pogodni su za korisnika koji su zainteresovani za numeričku analizu i brzu obradu manje količine informacija. Ne treba zaboraviti da se većina stonih računara koji normalno rade na bejziku može opremiti posebnim programom, paskal kompajlerom, pomoću koga će moći da koriste i drugi jezik. Takav program, doduše, oduzima nekih 16 kilobajta RAM-a, ali se sa takvim gubitkom lako mirimo ako nam za naše programe preostane još 16 kilobajta.

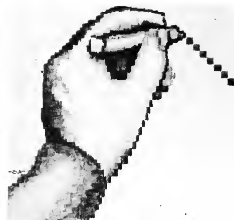
U poslednje vreme pojavljuju se računari koji koriste programski jezik forth. To je jezik nižeg nivoa od bejzika (što je jezik nižeg nivoa, to je razumljiviji za računar i manje pogodan za korisnika) koji nudi neke vrlo privlačne mogućnosti: definisanja novih naredbi, promene postojećih, znatno brže izvršavanje programa i slično. Na sreću, većina računara može da se opremi programom koji se zove forth compiler i tako raspolaže i bejzikom i fortom.

Ukoliko se odlučimo za nabavku nekog računara koji ne koristi bejzik, moramo da imamo u vidu da na specijalizovanim jezicima nije napisano mnogo programa. Vrlo je verovatno da ćemo sve programe koji nam trebaju morati sami da pišemo, što će značajno umanjiti vrednost našeg računara. Izgleda da, u sadašnjem trenutku, bejzik predstavlja najbolji izbor.

Tastatura

Tastatura je vrlo bitan element pri izboru računara. Kod većine kompjutera tasteri su raspoređeni po takozvanoj QWERTY shemi (kao na standardnoj pisaćoj mašini), a nekima od njih su dodeljene i posebne funkcije. Bolji računari, po pravilu, imaju veći broj tastera, dok slabiji računari imaju manju, nabijenu tastaturu na kojoj je svakom tasteru dodeljeno po nekoliko funkcija.

Prilično je broj korisnika koji pomišljaju da je najbolje kupiti računar sa takozvanom „senzorskom" tastaturom. Ova pomisao potiče od nesvesne analogije sa televizorima: televizor sa senzorima je obično bolji i skuplji od televizora sa mehaničkim ko-



Crtanje na ekranu. Grafička rezolucija računara se najdirektnije odražava na izgled slova i vernost slike

mandama. Kod računara je situacija upravo obrnuta — senzorska tastatura je jedna od najgorih karakteristika koju neki računar može da ima.

Za dobro i brzo kucanje je potrebna takozvana profesionalna tastatura sa mehaničkim tasterima. Na takvoj tastaturi dirke su normalne veličine, a pri pritiskanju se pomeraju kao na električnoj pisaćoj mašini. Svaki taster ima jednu ili dve funkcije (kada se zajedno sa njim pritisne i SHIFT, taster daje veliko slovo). Sve naredbe se kucaju slovo po slovo (npr. PRINT se kuca pritiskom na slova P,R,I,N i, konačno, T), a kod posebno dobrih računara postoji i grupa tastera kojima se mogu dodaliti neka češće korišćene naredbe radi bržeg kucanja.

Računari sa senzorskom tastaturom su znatno manji. Tasteri na njima se ne pomiču kada ih korisnik pritiska i, što je još gore, korisnik najčešće ne zna da li je dovoljno snažno pritisnuo taster da bi računar registrovao njegovu akciju. Na „senzorskim računarima" svakom tasteru je dodeljeno po nekoliko funkcija: umesto da kuca P R I N T korisnik samo pritiska taster P u pravom trenutku. Ovo na prvi pogled izgleda bolje, ali pola sata rada na „senzorskom računaru" će svakoga uveriti u suprotno. Pre svega, svaku naredbu treba naći na tastaturi (svaki taster ima po tri ili četiri dodeljene naredbe), zatim treba smisliti koje sve SHIFT dirke treba pritisnuti da bi se izabrala prava funkcija, a onda sve te dirke treba držati istovremeno i nadati se da će računar da registruje njihov pritisak. Za kucanje programa na senzorskoj tastaturi potrebno je, zavisno od iskustva korisnika, 2-20 puta više vremena nego kada se koristi profesionalna.

Prednost „senzorskih računara" je u tome što su znatno jeftiniji. Čini nam se da su, za većinu korisnika, njihove karakteristike ipak nezadovoljavajuće. Ukoliko se bavite nekim poslom u kome je potrebno dosta pisanja, pre ili kasnije ćete početi da koristite programe za obradu teksta. U tom slučaju zaboravite sve računare osim onih sa profesionalnim tastaturama! Ukoliko nameravate da sami pišete veći broj programa ili da kucate programe koji se objavljuju u časopisima, „senzorski računari" ponovo nisu najbolji izbor. Ali, ukoliko želite da koristite računar za igre (kupovinom komercijalnih programa), senzorska tastatura je prava stvar za vas. U većini igara operiše se sa svega nekoliko tastera i nema nikakve potrebe da plaćate skupu profesionalnu tastaturu.

Između senzorske i profesionalne nalazi se specijalizovana tastatura koju, na primer, koristi ZX Spectrum. Na ovoj tastaturi dirke se pomeraju kada ih pritiskate, pa imate osećaj da li ste pritisnuli neki taster ili ne. Pored toga, tastatura je i dalje malih dimenzija pa je svakom tasteru dodeljeno po nekoliko funkcija. Tastature poput ove su nepovoljne za kucanje dugih programa ali mogu da posluže za obradu teksta. Oni koji nameravaju da koriste računar za igre, vežbu programiranja i izvesnu količinu pisanja mogu da se opredele za računare kao što je ZX Spectrum. Ipak, ukoliko je novčani fond za kupovinu računara iole bogatiji, nije loše odlučiti se za model sa profesionalnom tastaturom.

Set karaktera

Neki računari poseduju jednostavan set karaktera koji se sastoji od velikih slova, brojeva i dvadesetak specijalnih znakova (tačka, zarez, dve tačke, upitnik, uzvičnik, zagrada, znak za dolar, navodnici i slično). Drugi računari mogu da koriste mala slova i gomilu specijalnih znakova računajući i većinu grčkih slova. Na korisniku je da odluči kakav mu set karaktera odgovara za





Važnije nego što izgleda kvalitet tastatura utiče na mogućnosti primene računara

ona koji žele da pišu programe, igraju se i uće programiranja mala slove i specijalni simboli su napotrabni. Za ona koji žela da im računar, pomoću štampača, održava prepisku, piše spiskove za nabavke i slično, mala slova su neophodna ali mnoštvo specijalnih simbola nije. Najzad, za one koji se bave pisanjem naučnih tekstove i knjiga kompletan set specijalnih simbola može da buda esencijalna karakteristika pri nabavci računara.

Neki računari poseduju mogućnost definisanja sopstvenih karaktera. Kod njih korisnik, pomoću posebnih naredbi, nekim tasterima može da dodeli nove znakove, kao što su naša slova š, č, ć i slično. Ovi računari obično spadaju u „višu klasu'' ali ih ne traba kupovati ukoliko nemate nameru da ih dopunite skupim štampačem. Nema nikakve koristi od ekrana na kome je prikazano slovo č ako ne raspolažemo štampačem koji to slovo može da prenese na papir!

Ekran

Većina računara se povezuje se standardnim televizorom crno-belim ili kolor. Neki računari poseduju tzv. monitorski izlaz koji omogućava povezivanje sa posebnim monitorom, a drugi imaju ugrađen monitor. U celini gledano, računari se ugrađenim monitorom nisu isplativa investicija: kupovina štampača se mnogo više isplati nego kupovina monitora.

Slika na monitoru je, po pravilu, bolja od slika koja se dobija na standardnom televizoru, ali razliku u većini primena možemo potpuno da zanemarimo. Važna karakteristika svakog računara je rezolucija. Pod rezolucijom podrazumevamo broj tačaka na ekranu koja računar može potpuno da kontroliše. Svaku od tih tačaka korisnik može, pomoću odgovarajuće naredbe, osvetliti ili zatamniti crtajući na taj način sliku na ekranu. Kod kolor računara korisnik može da bira i boju svake tačke ili boju segmenata na ekranu (svaka tačka se, u stranim tekstovima, naziva „pixel" pa se najčešće govori o „rezoluciju u pikselima'').

Jeftiniji računari imaju grafiku takozvana niske rezolucije. na ekranu može da se kontroliše svega oko 3000 tačaka, dok neki skuplji modeli mogu da adresiraju čak 256 000 tačaka. Negde na sredini se nalaze računari sa „pristojnom'' rezolucijom od oko 50000 tačaka na ekranu.

Na prvi pogled izgleda da je, ukoliko materijalne mogućnosti dopuštaju, dobro nabaviti računar koji ima što veću rezoluciju. Međutim, nije uvek tako: računar koji poseduje rezoluciju 640 256 i šesnaest boja, na primer, zahteva da 320 kilobajta RAM-a bude rezervisano za smeštanje ekrana. Ukoliko takav računar ima 330 kilobajta memorije, sve dobre karakteristike (visoka rezolucija i velika memorija) rezultiraju jednom lošom — samo 10 kilobajta preostaje za korisnikove programe i podatke.

Vrlo je bitno i koliko slova računar može da prikaže u jednom redu. Računari srednje klase koji rade u boji sa pristojnom rezolucijom najčešće u redu mogu da prikažu svega 32 karaktera, što je malo ako želite da ih koristite za obradu teksta. Treba imati u vidu da jedan kucani red ima između 60 i 70 znakova i da broj karaktera u redu manji od 64 zatvara mnoga područja primene računara. Oni koji žele da koriste računar u „ozbiljne'' svrhe (izuzetak su oni koji žele da ga koriste za crtanje vrlo



komplikovanih šema i planova) neće pogrešiti ako kupe računar koji radi sa crno-belim televizorom i ima 64 karaktera u redu. Oni koji žele da se igraju sa računarom će, naravno, odabrati neki „kolor kompjuter'', ali treba da, u granicama materijalnih mogućnosti, nastoje da broj karaktera u redu bude što je moguće veći — bar 40

Idealno bi bilo imati prilagodljiv računar korisniku bi mogao da bude ponuđen izbor između crno-bele ili kolor slike, velike i male rezolucije, broja karaktera u redu i slično Koliko nam je poznato, samo jedan računar iz srednje klase ima ove mogućnosti: to je BBC mikroračunar, posebno model B Korisnik, zavisno od potreba, može da menja broj adresivih tačaka i boja na ekranu, kao i broj slova u redu. Što je najlepše, u toku izvršavanja istog programa, ovaj broj može da se menja nekoliko puta — prema potrebama Jedini problem je u tome što su mnogi potencijalni vlasnici računara, sledeći naše uputstvo, već isključili BBC B iz izbora zbog cene od 400 funti.

Neki računari imaju ugrađene komande za crtanje krugova, mnogouglova, linija, pa čak i za pravljenje „crtanih filmova'' ipak nam se čini da ove mogućnosti ne bi trebalo da prevagnu pri izboru računara ako kompjuter ima dobru rezoluciju, gotovo svaki korisnik će biti u stanju da napiše program za crtanje kruga, animiranje slika i slično Ukoliko biramo između dva računara približne cene i karakteristika izabraćemo, jasno, onaj koji ima više specijalizovanih naredbi koje nam odgovaraju

Kasetofon

Bezmalo svi stoni računari omogućavaju snimanje programa i podataka na kasetofon Pošto je kasetofon uređaj koji je odvojen od računara, mogao bi da se smatra da on spada u periferijsku opremu No, čak i vlasnik računara koji ne namerava da nabavlje nikakve dodatne uređaje, svakako će osetiti potrebu da svoj računar poveže sa kasetofonom — zato su mogućnosti ovog uređaja obuhvaćene ovim poglavljem

Računar se sa kasetofonom povezuje pomoću kabla koji je obično deo standardne opreme kompjutera Na obične kasete se snimaju svi programi koji će korisniku docnije biti potrebni ili podaci koji su delimično obrađeni Ovo upisivanje, jasno, ne može da se obavi trenutno — računaru je potrebno određeno vreme da snimi neki program Brzina snimanja zavisi od računara (a ne od kasetofona kako neki početnici misle) i meri se u baudima Baud je bit u sekundi — termin koji početniku izgleda misteriozno No, sve postaje jasnije kada se pogleda nekoliko primera računar koji snima podatke brzinom od 1000 bauda, na primer, snimiće 16 kilobajta memorije za oko 2,5 minuta Ova brzina je sasvim prihvatljiva kada se učitavaju programi za igre ili za naučno-tehničke proračune, ali je neprijatna ukoliko računar treba da obrađuje veću količinu informacija koja je smeštena na kaseti

Neki računari imaju „potpunu kontrolu kasetofona'' Ovim terminom se obično označava činjenica da neki računari mogu da kontrolišu uključivanje i isključivanje kasetofona računar, na primer, najpre uključi kasetofon, učita program i na kraju zaustavi traku Kod učitavanja programa ova osobina nije mnogo važna, ali je za „ažuriranje datoteka'' presudna Šta je ažuriranje datoteka?

Neki korisnici planiraju poslovnu primenu računara Vlasnici objekata takozvane male privrede, na primer, često imaju potrebe za računarom koji bi organizovao nabavku repromaterijala, olakšao vođenje poslovnih knjiga i automatizovao kontakte sa mušterijama preko pošte Takvi korisnici, kako se to stručno kaže, koriste računar za rad sa datotekama i njihovo ažuriranje Datoteka je skup podataka i sastoji se od slogova Vlasnik biroa za fotokopiranje bi, na primer, mogao da oformi datoteku materijalnih troškova. Jedan slog ove datoteke bi obuhvato troškove nabavke papira, drugi održavanja uređaja, treći režijske troškove, četvrti plate službenika i slično Svakoga dana vlasnik unosi podatke o novim troškovima, a računar pronalazi potrebne slogove na kaseti, unosi ih u memoriju, menja njihov sadržaj i ponovo ih snima

Vlasnik biroa za foto-kopiranje koga smo pomenuli bi naišao na priličné probleme ukoliko bi, za pomenute poslove, koristio standardni kasetofon On bi morao neprekidno da premotava traku, traži početke slogova, kontroliše njihovo upisivanje u memoriju računara i ponovo premotava trake da bi se rezultati snimili na prava mesta Za ovakve potrebe nešto je pogodnije nabaviti računar koji se povezuje sa specijalizovanim kasetofonom Specijalizovani kasetofon, jasno, predstavlja dodatni trošak od pedesetak funti, ali zato automatizuje mnogobrojne radnje računar može sam da kontroliše premotavanje traka, upisivanje i brisanje podataka i slično. Obzirom da specijalizovani kasetofoni rade nešto brže od standardnih, oni predstavljaju dobru alternativu skupim disk-jedinicama (o disk-jedinicama ćemo govoriti u delu ovog izdanja koji je posvećen periferijskoj opremi) Ipak, za korisnike koji će upotrebljavati računar za igru, naučno-tehnička proračuna i vežbe programiranja, specijalizovani kasetofon je isto što i posebni monitor — nepotreban trošak

Ukoliko smo u prilici da kupujemo standardni kasetofon koji ćemo upotrebljavati uz računar (najčešće koristimo kasetofon koji već posedujemo), moramo da obratimo posebnu pažnju na neke sitnice. Treba pre svega, na svaki način da kupimo kasetofon koji poseduje brojač trake Nije posebno bitno da kasetofon bude posebno dobar (i



skup) — najobičniji mono-kasetofon će zadovoljiti sve naše potrebe. Dolazi, jasno, u obzir i upotreba stereo-uređaja, ali uz jednu ogradu: ukoliko pokušamo da čitamo kasete koje su snimane na mono uređajima pomoću stereo opreme, možemo da očekujemo teškoće jer će signali biti oslabljeni. Kako se većina komercijalnih programa prodaje na kasetama koje su snimane na najobičnijoj opremi, vlasnici stereo-uređaja možda neće biti u stanju da unesu kupljene programe u računar. Ovo ponekad ne predstavlje veliki problem (običan kasetofon se lako pozajmljuje od prijatelja), ali može da bude vrlo neprijatno. Šta da se radi ako je komercijalni program zaštićen pa ne može da se presnimi?

Poslednja sitnica na koju treba obratiti pažnju su priključci. Lako može da se dogodi da stari kasetofon i novi računar nemaju iste priključke za povezivanje. Ovoga se ne treba mnogo bojati: za stavljanje odgovarajućih priključaka nije potrebno nikakvo poznavanje elektronike — dovoljna je obična lemilica.

Generator tonova

Neki računari poseduju ugrađeni minijaturni zvučnik koji može da generiše tonove u nekoliko oktava. Neki kompjuteri imaju višekanalni ton-generator koji omogućava istovremeno generisanje nekoliko tonova (najčešće tri), drugi koriste zvučnik televizora kao izlaz, a treći nemaju mogućnost generisanja tonova.

Čini nam se da mogućnost generisanja tonova nije mnogo bitna. Korisno je da računar ima neku vrstu bipera kako bi upozorio korisnika na završetak nekog dugotrajnog posla, ali multiopcioni ton-generatori najčešće predstavljaju nešto što je prosečnom korisniku sasvim nepotrebno. Ozbiljnu primenu generatori tonova nalaze najčešće u igrama (na očajanje ukućana vlasnika računara) i sintezi glasa. Čak i na računarima sa skromnim mogućnostima generisanja tonova kao što je ZX Spectrum može da se sintetiše govor. Ipak, sinteza govora je područje koje zahteva priličnu stručnost korisnika, poznavanje akustike i dosta slobodnog vremena. Većina korisnika može samo da uživa u govoru svog računara za vreme nekih igara, pri čemu se koriste skupi programi. Sve u svemu, ugrađen generator tonova je lepa ali ne bitna osobina računara.

Isto se odnosi na ugrađeni časovnik. Neki računari imaju mogućnost pokazivanja vremena i datuma, ali je „doterivanje" časovnika i kalendara neophodno svaki put kada uključimo računar. Izgleda da je bolje kupiti časovnik i držati ga pored računara nego plaćati za dodatnu elektroniku. Jedna od retkih primena ugrađenih časovnika je merenje vremena razmišljanja u toku partije šaha.

Ugrađeni interfejsi

Za svaki računar može da se nabavi dodatna periferijska oprema u vidu štampača, disk jedinica, plotera i slično. Iako ćemo o periferijskoj opremi govoriti u okviru posebnog poglavlja, ovde ćemo se zadržati na mogućnostima njenog povezivanja sa računarom. Dva računara ili računar i periferijska oprema se povezuju pomoću takozvanog **interfejsa.** To je elektronski uređaj koji prilagođava signale koje šalje jedan računar tako da može da ih „razume" drugi, i obratno. U gotovo svaki računar je ugrađen interfejs za povezivanje sa kasetofonom i televizorom, a neki kompjuteri imaju i specijalne interfejse. Najpoznatiji od njih se obeležava sa RS232. Ukoliko u tabeli vidite da neki računar ima ugrađen RS232 interfejs, on može da se poveže sa obiljem periferijske opreme koju proizvode mnoge firme. Ukoliko planirate da docnije proširujete mogućnosti računara, ili želite odmah da nabavite neku periferijsku jedinicu (npr. štampač), neophodno je da konsultujete tabele i saznate da li se računar koji ste odabrali može direktno povezati sa štampačem koji vam je potreban ili je potrebna kupovina dodatnog interfejse. Za povezivanje sa disk-jedinicama bezmalo svaki računar iz srednje klase mora da ima poseban interfejs ili takozvani „disk-kontroler" (skup integrisanih kola koja kontrolišu disk-jedinice), dok se svi ostali periferijski uređaji najčešće povezuju sa računarom posredstvom RS232 interfejsa.

Predlažemo vam da, ako je ikako moguće, kupite računar koji ima ugrađen neki standardni interfejs (najbolje RS232) jer ćete se tako obezbediti za slučaj da u budućnosti poželite da proširite područje primene vašeg kompjutera.

Brzina rada

Brzina rada nekog računara zavisi od mnogobrojnih parametara: korišćenog procesora, frekvencije kvarc-kristala, kvaliteta sistemskog programa koji je ugrađen u ROM, jezika koji se koristi itd. Za korisnika svi ovi parametri nisu od naročitog značaja — on želi da njegov računar izvršava programe što je moguće brže.

Brzina rada je obično veća što je računar skuplji. U novije vreme u računare se ugrađuju takozvani šesnaestobitski procesori koji, pored ubrzavanja rada, povećavaju raspoloživu memoriju, unapređuju numeričku preciznost i olakšavaju pisanje mašinskih programa. No, računari sa šesnaestobitskim procesorima su i dalje prilično skupi, pa korisniku sa prosečno dubokim džepom preostaje samo da se opredeli za neki od osmobitnih modela. A njima je brzina prilično slaba tačka.

Sa aspekta potreba, brzina nije uvek bitna. Ako, na primer, koristimo računar za složene naučno-tehničke proračune, manja brzina rada može da se podnese — umesto da računar završi posao za 10 minuta, završiće ga za 20, što je i dalje nekoliko stotina puta kraće vreme u odnosu na ono koje bi nam bilo potrebno za manuelno rešavanje problema. Kod poslovnih primena brzina takođe nije veliki problem — obrada podataka je ovde relativno kratkotrajna i bitno je da računar može da memoriše dovoljnu količinu informacija. No, za igre je brzina esencijalna: ukoliko računar radi sporo, nećemo moći da napišemo programe za dinamične igre.

UPOREDNE KARAKTERISTIKE STONIH RAČUNARA

Naziv	Cena (u funtama)	Procesor, clock	RAM	Proširenje RAM-a	ROM	Proširenje ROM-a	Programski jezici	Dodatni programski jezici	Tastatura	Set karaktera
Apple II	700	6502 1 MHz	48 K	64 K	12 K	expansion kartice	BASIC	Većina postojećih (FORTRAN, PASCAL, COBOL, Pilot itd)	profesionalna	velika slova
Apple III	2416	6502 2 MHz	128 K	256 K	12 K	kartice	BASIC	Pascal Cobol asembler	profesionalna	velika/mala definisanja karaktera
Atari 400	200	6502 1.8 MHz	16 K	nema	10 K	Moduli	Nema (potreban modul)	BASIC Asembler Pilot, FORTH PASCAL	Senzorska, 61 taster	velika/mala
Atari 800	450	6502, 1.8 MHz	16 K	48 K	10 K	moduli	Nema	kao Atari 400	profesionalna, 61 taster	velika/mala
Atom Electron	199	6502, 2MHz	32 K	32 K	32 K	32 K	BASIC	nema	profesionalna 56 tastera	velika/mala
BBC micro model B	399	6502, 2MHz	32K	32 K	32 K	Moduli	BASIC asembler	FORTH	Profesionalna, 72 tastera	velika/mala, definisanje karaktera
Color Genie	200	Z80, 2.2 MHz	16K	32K	16 K	moduli	BASIC	nema	loša, 56 tastera.	velika/mala, 128 grafičkih i 128 programabilnih ka/aktera
Commodore 64	300	6510, 1MHz	64 K	nema	20 K	Moduli	PET BASIC	PASCAL, FORTH	profesionalna 66 tastera	velika/mala, specijalni znaci
Commodore 4016	445	6502, 1MHz	16 K	32 K	16 K	nema	BASIC	uskoro	profesionalna 74 tastera	velika/mala, grafički karakteri
Commodore 8032	985	6502, 1 MHz	96 K	nema	16 K	moduli	BASIC	većina postojećih	Profesionalna 73 tastera	velika/mala
DAI PC	664	8080		nema	24 K	nema	BASIC	uskoro	profesionalna	velika/mala
Datac MC	430	Z80 2MHz	16K	nema	12K	16 K	BASIC	Asembler	Prof 57 tastera	velika/mala, grafički
Dragon 32	199	6509E	32 K	64 K	16 K	moduli	BASIC	nema	poluprof 53 tastera	velika slova
Genie I i II	345	Z80, 1.7 MHz	16 K	46 K	13 K	nema	BASIC	razni	profesionalna, 71 taster	velika/mala
Hewlett-Packard 86	1617	specijalizovani	16K	120 K	32 K	nema	BASIC	nema	visoko prof	velika/mala, specijalni
Hewlett-Packard 87	2400	specijalizovani	128 K	640 K	48 K	kartice, moduli	BASIC	nema	visoko profesionalna	velika/mala, specijalni znaci
IBM PC	3395	8088, 4.7 MHz	64 K	256K	40 K	nema	BASIC	Pascal	profesionalna, 93 tastera	velika/mala, spec. znaci
Nascom 2	327	Z80 4MHz	2K	48 K	16 K	nema	BASIC	PASCAL	profesional	velika/mala
Newbrain A	229	Z80a 4MHz	32 K	288 K	29 K	moduli	BASIC kompajler	nema	poluprof. 62 tastera	512 karaktera računajući grčka slova
Sinclair ZX81	40	Z80	1K	16K/64K	6 K	nema	BASIC	FORTH asembler	senzorska 40 tastera	velika slova
Sinclair Spectrum	100	Z80, 3.5 MHz	16 K	46 K	16 K	moduli	BASIC	PASCAL, FORTH, asembler	neprofesionalna, 40 tastera	velika/mala
TRS 80 I	289	Z80 2MHz	4 K	46 K	12 K	nema	BASIC	FORTAN, LEV3 BASIC Pascal	profesionalna	velika
TRS 80 kolor	230	6809, 1 MHz	4K	32K	6 K	da	BASIC	extended color BASIC	poluprof 53 tastera	velika
TI99/4A	150	9900	16K	48 K	26 K	moduli	BASIC	Extended BASIC LOGO Pascal Asembler	profesionalna	velika/mala
VIC 20	164	6502 1MHz	5K	29 K	20 K	nema	BASIC	FORTH	64 tastera	velika/mala grafički

Ekran	Slova na ekranu	Grafika u tačkama	Kasetofon	Priključenje diskova	Zvuk	Ugrađeni interfejsi	Adresa proizvođača	igre naučna primena poslovna primena programiranje ukupni rejting
ili monitor	40 24	do 256 192	svaki	uključen flopi 140 K	da	Portovi omogućavaju povezivanje se većinom posloječe opreme	Apple Comp Finlay Road Hemel Hempstead Herts HP2 7PS	77–65–67–60–67
B monitor	80 24	promenljiva max 560 192	nema	1 140 K flopi uključen	da	kao APPLE II	kao Apple II	72–80–85–70–71
olor TV	40 24	320 192	Sopstveni 50 funti	flopi do 88 K	četiri kanala	Dokupljuje se RS232 (135 funti)	Atari International 185 Ealing Road Alperton Middlesex	67 28–20–40–43
olor TV	40 24	320 192	Sopstveni 50 funti	flopi do 88 K	kao Atari 400	kao Atari 400	kao Atari 400	62–38–35–44–52
r TV, monitor	max 80 24	promenljiva max 640 256	svaki	flopi	1 kanal	kroz dodatke	Acorn Computers, 4a Market Hill Cambridge CB2 3NJ	75–40–35–80–58
olor TV ili B monitor	max 80 30	promenljivo max 640 256	svaki	flpol	3 kanala	RS423 Prestel I/O port 4 kanala	Acorn Computers 4a Market Hill Cambridge	80–55–60–85–70
/ monitor	24 40	160 88	svaki	flopi	preko TV	RC232C EG2014 EG2013	Lowe Electronics Chestefield Road Matlock Derbyshire DE4SLE	75–43–22–38–45
olor TV	40 25	do 320 200	Sopstveni 45 funti	VIC1541 flopi	da	RS232	Commoadre 657 Ajax Ave Slough Berks	70-35 36 49 48
nonitor	40 25	40 25	sopstveni	PET 4040 flopi	nema	1EEE 488 bus	Kao Commodore 64	35–58–65–50–51
nonitor	80 25	80 25	sopstveni	moguća dogradnja flopi i pravih diskova	nema	1EEE-488 bus	kao Commodore 64	42–50–45–45–46
olor TV	60 24	promenljiva 256 335	dva kasetofona	ugrađen disk kontroler 2 flopi po 800 K.	četiri kanala	RS232	Data Applications 168 Dyer St Cirencester Glos GL7 2PF	53–50–45–48–49
TV	40 24	40 24	svaki	nemoguće	nema	RS232	Datac Tudor Road Altrincham Cheshire	36–40–75–35–46
monitor	32 16	64 32	svaki	nema	kroz TV	Centronix Port Expansion port	Dragon Data Queensway Swansea Industrial Estate, Swansea SA54EH	68–32–20–46–44
monitor	64 16	128 48	svaki	samo uz dodatke	nema	nema	Lowe Electronics Chestfield Road Matlock Derbyshire	58–40–40–60–48
den monitor	32 16	192 256	specijalni ugrađen	dogradnja 2 flopija	da	RS232 HP-IB HP-IL	Hewlett-Packard 1000 N E Circle Boulevard Corvallis Oregon 97330 USA	25–55–72–70–56
den monitor	64 20	velika	nema	priključivanje flopi i pravih diskova	da	HP-IB HP-IL, RS 232	kao HP85	27 56 70-56 59
den crno-beli monitor	80 25	visoka	nema	ugrađena dva flopi po 160 K	da	RS 232 i drugi	KGM Micros 14 Windsor Rd Slough Berks SLI 2EJ	33-76–59–62–57
monitor	16 48	48 96	svaki	flopi	nema	RC232C	Lucas Logic Ltd Warwick CV34 5P7	38–50–54–48–48
TV	80 24	veliks	svaki	nema	nema	RS232 povezivanje sa štampačem	Više se ne proizvodi u GB	35–40–40–35–37
TV	32 24	64 48	svaki	nema	nema	nema	Sinclair Research 6 King Parade, Cambridge CB2 1SN	46–16–10–18–22
olor TV	30 24	256 176	svaki	Sinclair Microdrive (nije pravi disk)	da	RS232 košta 30 funti	kao ZX81	62–23–24–33–35
monitor	64 16	128 48	svaki	flopi uz Expansion Interface	nema	Expansion interface dokupljuje se	Tandy Corporation. Tameway Tower Bridge St Walsall West Midlands WS1 1LA	68–49–70–49–59
olor TV	32 16	256 192	svaki	flopi	da	RS232 portovi	Kao TRS 80	74–38–30–46–47
olor TV	32 24	32 24	svaki	flopi	3 kanala	Port za palice expansion box RC232 i drugi	Taxas Instruments Manton Lane Bedford MK41 7PU	45–30–40–55–43
monitor	22 93	do 176 158	sopstveni	flopi 170 K	3 kanala	port expansion	kao Commodore 64	80–20–25–30–39

Kako da procenimo brzinu računara? Najbolji pregled se stiče kada napišemo program koji rešava neki složen matematički problem i primenimo ga na nekoliko kompjutera. No, malo nas ima priliku da istovremeno radi sa nekoliko računara, pa smo najčešće upućeni na testove koji se objavljuju u časopisima. U tim testovima računaru se postavljaju različiti problemi i meri se vreme potrebno za njihovo rešavanje. Zatim se vremena potrebna za rešavanje pojedinih problema ispišu u tabelu, nađe se njihova srednja vrednost i računari se sortiraju prema njoj. Tako nastaje tabela testova brzine ili, kako ih nazivaju, „benchmarks" testova.

Jedna takva tabela data je i u ovom tekstu. Nismo, jasno, mogli da obuhvatimo preveliki broj računara, ali nam se čini da su među njima i mnogi popularni modeli. U tabeli se nalaze stoni i džepni računari, površan pogled pokazuje da su novi džepni računari često značajno brži od nekih stonih modela dok se stari programabilni kalkulatori nalaze na samom začelju liste.

Testovi brzine

Model	BM1	BM2	BM3	BM4	BM5	BM6	BM7	BM8	prosečna brzina
TI Professional	1.0	4.2	9.3	9.7	10.5	19.0	29.5	31.0	14.3
BBC Micro	1.0	3.1	8.2	8.7	9.1	13.6	21.4	51.0	14.6
Acorn Electron	1.1	4.0	11.1	11.8	12.4	16.7	28.7	72.5	20.0
Camputers Lynx	1.7	4.3	12.4	8.9	10.4	16.3	29.8	86.6	21.3
Torch (BBC Basic)	1.4	5.4	14.4	15.3	16.1	24.8	38.4	89.0	25.6
Epson QX-10	2.3	6.4	16.8	15.8	16.5	31.9	52.9	65.9	25.9
Newbrain	2.0	5.6	19.2	17.5	19.2	32.0	48.8	70.0	26.8
BM Personal Computer	1.5	5.2	12.1	12.6	13.6	23.6	37.4	35.0	17.6
Commodore VIC 20	1.4	6.3	15.5	17.1	16.3	27.2	42.7	99.0	28.7
Appla II	1.3	6.5	16.0	17.8	18.1	28.6	44.8	107.0	30.4
Dragon 32	1.6	10.2	18.7	21.8	23.3	34.3	50.0	129.0	36.2
Hewlett Packard HP86	3.0	5.2	19.4	18.8	20.4	36.5	56.5	134.0	38.7
Colour Genie	2.7	10.6	25.0	25.8	28.9	47.6	73.1	104.3	39.8
Tandy TRS-80 Color Computer	2.0	11.3	22.2	23.9	27.0	41.6	61.1	130.0	39.9
Tandy TRS-80 Modal I Level II	2.7	11.6	28.0	28.5	31.3	51.9	91.0	117.0	44.0
Video Genia	2.7	11.6	28.0	28.5	31.3	51.8	81.0	117.0	44.0
Sinclair ZX81 (fast moda)	4.6	6.9	16.4	15.8	18.6	49.7	68.5	229.0	51.2
Epson HX-20	2.7	15.3	33.1	32.8	35.3	59.1	100.6	133.3	51.5
Sinclair Spectrum	4.8	9.7	21.1	20.4	24.0	55.3	80.7	253.0	58.5
Onc 1	2.0	17.3	29.4	31.7	38.1	50.1	76.1	233.4	59.8
Tandy TRS-80 Model 100	3.6	9.5	26.6	28.8	31.5	43.0	64.0	321.0	66.1
Taxas TI99/4a (standard)	3.0	9.0	24.0	24.8	26.2	61.6	84.6	384.0	77.2
Taxas TI99/4A (extended)	6.5	18.5	40.0	40.1	42.0	98.4	140.3	240.0	78.2
Casio PB-100	8.0	39.0	92.0	80.0	105.0	160.0	220.0	341.0	128.4
Sharp PC1500	15.0	70.0	121.0	122.0	178.0	293.0	383.0	510.0	211.5
Sharp PC-1251	42.3	70.6	162.5	165.8	197.3	427.8	581.4	980.0	328.6

Programska podrška

Iako programsku podršku pominjamo tek na kraju ovog poglavlja, ona može da bude bitnija od svih pobrojanih osobina. Novi vlasnik računara ne može da očekuje da će ući u sve tajne programiranja u kratkom roku. Zato će, ukoliko želi da koristi računar ili da se igra sa njim, biti prinuđen da kupuje programe koje su drugi sastavili. Pored toga, čak i vrlo iskusan korisnik računara pre ili posle dolazi do saznanja da se ne isplati pisati sve programe koji su mu potrebni. Svako voli da igra šah protiv računara, ali pisanje programa za šah zahteva konsultaciju obimne literature do koje je teško doći, utrošak nekoliko meseci za njegovo pisanje i možda i cela godine za testiranje i prepravke. Zar nije jednostavnije dati desetak funti za program za šah i početi zabavu bez ikakvog uloženog napora?

Primena računara za obradu teksta zahteva poseban program nazvan tekst-editor koji je napisao tim profesionalaca i oprobalo nekoliko stotina ljudi pre nego što je komercijalno počeo da se prodaje. Red sa datotekama zahteva posebne programe za njihovo sortiranje i obradu. Kao što ne možemo sami da projektujemo računar, tako ne možemo da napišemo ni sve programe koji će nam zatrebati. Zato smo upućeni na programe koji postoje na tržištu. Neki računari su, voljom proizvođača, namenjeni onima koji žele da se igraju, pa su u prodaji mahom programi za igre. Neki drugi računari, pre svega zbog dizajna tastature, nisu pogodni za obradu teksta pa je broj i kvalitet tekst-editora za njih minimalan.

Da bismo bar približno dali predstavu o mogućnostima primene pojedinih računara u nekim oblastima, u tabeli je data neka vrsta rejtinga. To je broj bodova od 1 do 100 koji sumira ugrađene karakteristike računara i raspoložive programe. Jasno je da brojke predstavljaju subjektivan utisak autora zasnovan na tabelama iz engleskog časopisa „What Micro?" i da je moguće da neke od njih potpuno ne odgovaraju realnosti. Uz sav rizik da budemo subjektivni na kraju svakog reda tabele smo sabrali četiri poslednje rubrike i rezultat podelili sa četiri. Na taj način smo dobili neku vrstu sumarne ocene svakog računara koja bi mogla da posluži za njihovo rangiranje. Računari su u tabeli ipak poređani po abecednom redu, a ne po našem „rejtingu", ali će vam poslednja kolona, nadamo se, pomoći da izaberete najbolji.

Konačan izbor

Čitajući tekst i konsultujući tabelu verovatno ste već precrtali računare koji sigurno ne ulaze u izbor. Tako ste došli do dva ili tri modela između kojih se treba odlučiti. Poslednja odluka je najteža. Ukoliko sami putujete u inostranstvo radi nabavke računara, ne opredeljujte se sve dok na delu ne vidite sve kandidate. Prodavci računara na zapadu su vrlo ljubazni i rado će vam demonstrirati mogućnosti svih računara koje želite da pogledate i odgovoriti na vaša pitanja. Pošto je njihov cilj da prodaju računar, jasno je da će neki odgovori biti „ulepšani", ali ćete, u suštini, dobiti informacije koje će biti pouzdane. Na osnovu njih trebalo bi da se odlučite i izvršite kupovinu.

Problem nastaje kada treba da se odlučite za računar koji će, umesto vas, kupiti neko drugi. Trebalo bi da posetite neki kompjuterski klub i da se raspitate za nekoga Jugoslovena koji poseduje računar koji želite da kupite. Ljubitelji računara su obično solidarni: vlasnik nekog računara će se rado odazvati vašoj molbi da ga posetite i na licu mesta pogledate kompjuter. Ukoliko ne možete da pronađete ni jednog vlasnika, raspitajte se među iskusnijim članovima kompjuterskog kluba koji prate strane časopise da li je o tom računaru dosta pisano, ima li njegovih karakteristika koje su posebno dobre ili posebno loše, da li je neko od njih naišao na neki program za taj računar i slično. U svakom kompjuterskom klubu će se verovatno naći neka „živa enciklopedija" koja će moći da vam pomogne pri izboru računara.

Postoje i neke „tajne" kojima treba da ovladaju potencijalni jugoslovenski vlasnici računara. U principu je dobro kupiti računar koji poseduje veći broj Jugoslovena. Na taj način se do stranih programa može doći mnogo jeftinije: nekoliko vlasnika računara se udruže, kupe program i prekopiraju ga u nekoliko primeraka. Pored toga, u okviru „malih oglasa" se stalno pojavljuju preprodavci stranih programa od kojih se kaseta vredna tridesetak funti može kupiti za 800–900 dinara, pa čak i jeftinije. Jasno je da se ovde radi o „kompjuterskom piratstvu", ali se moramo pomiriti sa tim da je kod nas uzeo maha. Najveći broj vlasnika računara u Jugoslaviji (barem prema našim podacima) poseduje ZX Spectrum i ZX 81. Popularni su, pored toga, TRS 80 model 1 (koji se, na žalost, zvanično više ne proizvodi iako može da se pronađe u nekim prodavnicama u Nemačkoj) i Apple II.

Već smo rekli da je vrlo važno izabrati računar koji odgovara vašim potrebama. Jugoslovenski vlasnici se, međutim, nalaze u nešto povoljnijem položaju od stranaca: ukoliko uvide da su napravili pogrešan izbor (na greškama se uči) mogu bez ikakvih problema da prodaju računar po otprilike dvostrukoj ceni! Kod nas je ponuda mala, a potražnja za boljim modelima ogromna.

Najpopularniji modeli računara

zvezde koje ne tamne

Prethodno poglavlje je pokušalo da objasni neke osnovne karakteristika računara i pomogne vam da suzite izbor ne nakoliko modela koji odgovaraju vašim mogućnostima i potrebama. Dafinitivan izbor, međutim, ne bi trebalo izvršiti pre nego što se pogledaju svi „kandidati“ izbliza i upozneju njihove karakteristike. Jedan od načina de to učinite je i da pročitate prikaze pojadinih modela koje ovde dajemo. Počećemo od jeftinijih modela koji su, koliko nam je poznato, popularni medu Jugoslovenima, a završiti sa nekoliko računara iz više klase koji to, verovatno, nikada naće postati. No, i ovi računari su neobično značajni za korisnika koji planiraju ozbiljnu poslovnu i naučno-tehničku primenu računara i kojima nije mnogo bitno koliko će novca izdvojiti za njege.

ZX 81

Klajv Sinklar, poznati angleski naučnik i biznismen, okušao se u mnogobrojnim dalatnostima i na tržište izneo mnogobrojne elektronske uredaje. Nisu mu uvek cvetale ruže: njagova firma Sinclair Research se nekoliko puta nalazila na samoj granici bankrotstva i bila spasavana eamo intervencijama engleske vlade. No, sredinom 1980. godine „Ujka Klajv“ (kako ga iz milošta zovu Englezi) ja na tržišta izneo proizvod koji će mu donati ogromnu popularnost i, zašto to ne reći, milione funti — računar ZX 80. Pri projektovanju ovog računara Sinklar se opredelio da po svaku cenu proizvede što jeftiniju napravu. Sva rešenja koja su bila skupa automatski su smatrana lošim rešenjima i nastao je ZX 80.

Iako moramo da kažemo da ja ZX 80 bio vrlo loš računar, njegova cana od pedesetak funti navela ja mnoge koji uopšte nisu pomišljali na nabavku računara da „progutaju udicu“. Za njih su karakteristike ZX 80 bile dobre jer druge računara nisu ni poznavali. No, kako eu napredovali, tako im je ZX80 postajao nedovoljan i oni bi, da Sinklar nija nešto učinio, verovatno prešli na neki malo skuplji model drugog proizvođača. Sinkler ja dobro proosnio tržište — u prodavnicama se godinu dana docnije pojavio ZX 81.

ZX81 predstavlja logični nastavak ZX 80, ali uz jednu bitnu razliku — računer ima dalako povoljnije kerakteristike. Pre svega, u ROM računara ja ugradan znatno bolji bejzik. Ovaj bejzik nija preuzet ni od jadne postojeće firma, već ja nezavisno razvijen, što ja imalo i dobre i loše strana. Dobra strane su se odrazile kroz canu računara, a loše kroz činjenicu da se programi piseni za ranije modele (Appla II, TRS 80 ...) mogu koristiti na ZX81 tek uz znatne prepravke. To Sinklara nije mnogo bnnulo: računar koji košta pedesetak funti će zainteresovati mnoge korisnike, oni će pisati programa za njega pa će i nezavisne firme vidati ogroman intares u preprodaji tih programa. Pokazalo se da je ovakvo mišljanje bilo „pogodak u centar“.

Bejzik računara ZX81 moža da se označi kao „vrlo dobar“. Korisnik može da ostvari sve programe koji su mu potrabni, od igara do naučno-tahničkih proračuna. No, dao ROM-a koji omogućava komforan rad sa računarom je prilično traljavo urađen. Pre svega, naredba ne mogu da se kucaju slovo po slovo (PRINT se, na primer, na moža kucati uzastopnim pritiecima na P, R, I, N i T) već preko tastera sa unapred dodaljenom funkcijom. Osim toga, naredba koje korisnik kuca se smeštaju u zadnja dva reda ekrana i računer iepieuja čitav tekst tekuće naredbe kada god ja konsnik dopuni jednim slovom, što nija samo dosadno već i zamorno za oči.

Drugi veliki nedostetak ZX81 ja što ja prilično spor. To, zasigurno, nije krivica procesora (Z80A), već loše urađenog bejzik interpretatora. Od kvaliteta ovog interpretatora, naima, zavisi brzina kojom računar pravodi svaku naredbu u oblik pogodan za njano izvršavanja, što sa neposredno odražava na brzinu rada. Pošto se generisanja karaktera na akranu odvija pod kontrolom procesora, uvedeni su FAST i SLOW modovi. U prvom se na akranu u toku rada na prikazuje nikakva slika, ali računar radi brža (to „brže“ i dalja nija mnogo brzo), dok se u SLOW modu slika naprakidno generiše što se negativno održave na brzinu rada. Postojala su, verovatno, i bolja rešenja, ali ja u 8 K ROM-a zaista bilo teško ugraditi i ovoliko.

Mamorija ZX81 je dovoljna za one kojima ja ovaj računar namenjen. Osnovna mamorija od svega 1 K verovatno služi samo da navede nekoga na kupovinu računara — čim ga iola bolje upozna shvatiće da u ovaj kilobajt ne može da stane ni čitav sadržaj akrana i požeieće da kupi modul za proširenje od 16 K. Šesnaest kilobajta je dosta mamonje — taman onoliko koliko ovaj računar može da opsluži. Docnija su se pojavili dodaci koji proširuju mamonju ZX81 do impozantnih 58 kilobajta, ali nam se čini da kupovina ovakvog modula predstavlja lošu investiciju. Računaru ja potrebno nekih 30 minuta da programom se traka popuni ovoliku mamoriju. Šesnaest kilobajta je dovoljno za igra i manja proračune a treba imati u vidu i činjanicu da su praktično svi komarcijalni programi namanjani ZX81 sa 16 K.

Računar koji hrabro nosi svoja mane: ZX81

Glavne zamerka se upućuju tastaturi. Eminentni „Byte“ piše da ja „ZX81 namanjan samoubicama koji žela da umiru postepeno pokušavajući da nataraju računar da ragistruja pritisek na neki taster“. Tasteri se, naima, ne pomeraju i korisnik nema osećaj da li ja naki pritisnut ili ne. Jedini način da to proveri ja da neprakidno prenoei pogled sa tastature na ekran i da ispravlja greške pomoću dirke DELETE (brisanja poslednjeg pritisnutog karaktera), koja ja, da stvar buda još gora, „šiftovane funckija“ (potrabno je istovramano držati pritisnuta tastera SHFT i DELETE da bi se poslednji karakter obrisao). Ovakva testatura ja dobra jedino za igra (u tom slučaju treba pritiskati samo par različitih tastera), mada se i tu javljaju problemi zbog brzog popuštanja nekih kontakata. Kucanja programa ja prilično frustrirajuća vežba koja, na neki način, destimuliše kreativan rad korisnika.

ZX81, ipak, hrabro nosi svoje mane. Mnogobrojni „ZX-ovci“ su uspeli da se naviknu na njega, zavola ga i opreme hiljedama programa. Pravo ja čudo kakve sve složene igra mogu da budu smeštene u magičnih 18 K. ZX ja možda spor, ali se pokazalo da ja njegov mašinski jazik dovoljno brz za sve vrste akcionih igara, eimulatora leta, šahovekih programa itd. Rezolucija ja, možda, mala ali će vešt programar na akranu prikazati vrlo složan trodimanzionalni lavirint sa monstrumom koji čeka iza nekog ćoška. Intaresantno ja da su nejbolji ZX programari regrutovani iz redova onih koji su se već okušali na džepnim računanma — ovi korienici su navikli na izvlačenja maksimuma iz skromnog hardvera.

I pored svaga, ne čini nam sa da ja danas pametno kupiti ZX81. Njegova cena (ispod 40 funti) je zaista niska, ali se za samo malo više para moža nabaviti nauporedivo bolja mašina. Ta mašina se zove ZX Spectrum.

ZX Spectrum

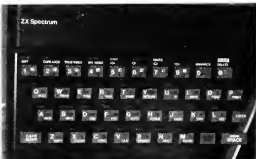
Verovatno je Sinkler procenio da reč Spectrum mnogo bolje zvuči nego ZX82 (možda se i bojao da će neiskusni kupci početi da mešaju ZX80, ZX81, I ZX82), pa je njegov sledeći kompjuter dobio baš ovo ime. Spectrum je, naime, vrlo ponosan na činjanicu da predstavlja prvi računar iz ekonomske klase koji poseduje grafiku takozvane visoke rezolucije: pod programskom kontrolom na ekranu sa može cevetliti svaka od 45056 tačaka. Osim toga, Spectrum može da prikazuje slike u više boja, iako korisniku nija ostavljena mogućnost da kontroliše boju svake tačke — grupe tačaka moraju da budu cevatljene istom bojom. U računar je ugraden časovnik i generator tonove (možda bi se on bolje okarakterisao kao biper obzirom da su tonovi koje proizvodi prilično slični i piskavi), a svoje mesto su pronašla i mala slova.

Spectrum ima ROM od 16 kilobajta, što znači da ja opremljan vrlo dobrim bejzikom. Ovaj računar ima mnogobrojne grafičke naredbe, koje su ranije vidane samo kod mnogostruko skupljih profesionalnih sistema (naredba CIRCLE, koja na ekranu crta krug, vidana je prvi put kod čuvenog HP45C). Naredba koja se kuca i dalje se nalazi u poslednjih par redova ekrana, ali ja ispravljanje daleko brže i komfornije. Ne postoja više FAST i SLOW mod — računar neprekidno prikazuje sliku, radeći otprilike onoliko brzo (sporo?) koliko i ZX81 u FAST modu. Brzina rada je, dakla, i dalje mala, ali je prihvatljiva, pogotovu kada se pišu programi na mašinskom jeziku.

Tastatura Spectruma pradstavlja znatno poboljšanje tastatura ZX81. Svakom tasteru je ponovo dodeljeno mnoštvo funkcija, ali su tasteri više razmaknuti I, što je najvažnije, kreću se kada ih korisnik pritisne. Tastatura je, jasno, dalaka od profesionalne, ali je sasvim prihvatljiva za kucanje dužih programa. U početku, doduše, sve ja vrlo komplikovano, s obzirom da svaku naredbu treba pronaći I smisliti koji sva SHIFT tasteri traba da se pritisnu da bi ona bila prepoznata. No, posle izvesnog vramena, korisnik nauči gde je koja naredba i kucanje postaja srazmarno brzo. Za igre ja tastatura odlična mada se kod nekih primeraka tasteri habaju posle dužeg pritiskanja i računer počinje teže da razaznaje neka slova.

Spectrum se prodaje u dve verzije: sa 16 K i sa 48K. Razlika između ove dve verzija se odražava kroz cenu: Spectrum sa 16 K košta 100, a sa 48 K — 130 funti. Već smo rekli da ja 16 K dosta mamorija, ali bismo vam preporučili da nabavite „prošireni Spectrum". Većina izvanrednih komercijalnih programa pisana je za ovu verziju. Osim toga, od prvih 16 K RAM-a nekih 9 ode na smeštanja ekrana i sistemskih variabli, pa korisniku ostaje samo 7 kilobajta.

Iako skuplji od ZX81, Spectrum se odlično prodaja. Ponovo je širok krug korisnika obezbedio ogromnu programsku podršku. Stručni časopisi su prapuni reklama kojima desetine kompanija reklamiraju na stotine programa. Ako smo rekli da su kod ZX81 najčešće akcione igre, za Spectruma napisano ja najviše igara-avantura. To je, verovatno, posledice veće mamorija (u 48 K može da stane priličan broj lavirinata, soba, zlih i dobrih duhova, stepenice ...) i skromne brzina rada.



Najbolji izbor u ekonomskoj klasi: ZX Spectrum

U doba kad ovo pišemo (septembar 1983.) Čuju se glasovi da Sinkler priprema nešto novo. Moguće je da će, kada ovo izdanje doda na kioske, to „novo" biti već par meseci staro, pa nagadanja nije mnogo zahvalno. Novi modal bi, navodno, trebao da se zove ZX83 i da predstavlja nadgradnju ZX81. Moguće je da će Sinkler odustati od kolor grafike (i tako održati proizvodnju i prodaju Spectruma na nivou koji je blizak sadašnjem), ali ća verovatno insistirati na visokoj rezoluciji. Tastatura će, po prilici, biti slična onoj koju poseduje Spectrum, iako će možda biti bolje uradena. Glavni napredak se očekuja kod brzine rada i (eventualno) poboljšanja bejzika. Cena bi morala da ostane niska (ispod 150 funti), a postoji čak i mogućnost da Sinklar odluči da u novi računar ugradi akran (zli jezici kažu da je to jedini način da mu se projaktovanje „mini-talavizora", na koje ja utrošio znatna sredstva, isplati). Ima i onih koji smatraju da Sinkler neće uskoro proizvesti novi računar pošto je ostao baz dvojice stručnjaka koji eu obavili glavni posao pri projaktovanju I programiranju ZX81 i Spectruma. Ova dvojice, Richard Altwasser i Steven Vickers, osnoveli su sopstvenu firmu koja proizvodi računar Jupiter Ace.

ZX računari se prodaju I u Sjedinjenim Državama i to pod imenom Timex-Sinclair. Iako su pratrpeli izvesne modifikacija za ovu priliku (prilagodeni su američkim talavizorima i dobili našto veću bazičnu memoriju) slabo se prodaju — izgleda da je njihove cena za Amerikance destimulišuće mala i da oni više vola da potroše više novca i dobiju bolji sistem.

Acorn i BBC

Engleska firma Acorn ja počela nedavno da proizvodi računare — njihov prvi uspeo modal je Atom Acorn. Ovaj računer košta svega 174 funta, ali sa za taj novac malo dobija: RAM od 2 kilobajta ja nedovoljan za bilo kakve primane, a u ROM-u se nelazi polovičan BASIC bez matematičkih funkcija, mogućnosti rada sa razlomljenim brojevima i slično. Da bi se računar doveo u funkcionalno stanja, treba kupiti dodatne RAM module (bar 12 K) i 2 ROM-a od po 4 K. Posla ovih proširenja Atom postaja pristojan mini-kompjuter iz akonomske klase: tastatura mu je profesionalna, poseduja velika i mala slova, koristi standardni talavizor u boji (rezolucija 256 x 192) i povezuje se se standardnim kasetofonom. Mala mamorija čini da se loša veza sa kasetofonom (samo 300 bauda) malo primećuja, a postoji mogućnost priključivanja disk-jedinice koja koštaju celih 350 funti. Sve u svamu, Atom je ralativno dobar računer, ali nije naročito praporučljivo kupovati ga — bolje ja da se opredelita za neki od BBC modela.

BBC modala A i B je takoda proizvela firma Acorn i to u sklopu obrazovnog programa angleske talevizija. Oba modala spolja gledano izgledaju isto — karakteriše ih lep dizajn, izvanredna tastatura i deset specijalnih tastera kojima korisnik može da dodeljuja grupe naredbi ili svoja programe. ROM oba računara ima po 32 K i u njemu ja smašten izvanredan bejzik — verovatno bolji od jezika koji poseduju svi pobrojani modeli. Razlog za to ja činjanica da je računar potpuno epreman za strukturalno programiranja.

O strukturalnom programiranju u okviru ovog izdanja nije bilo mnogo reči, iako je ono u poslednje vrema postalo neobično popularno. Idaja je u toma da se program piše u dalovima: izdvajaju se nezavlana celina koja obavljaju neku funkciju i pišu posebni programi za njih. Zatim se ti programi odvojeno testiraju i ispravljaju da bi se na kraju sastavio samo kratak glavni program koji povezuje njihov rad. Ovako pisani programi troše više mamorija od standardnih ali se daleko lakše pišu, testiraju, ispravljaju i prilagodavaju različitim računarima. Bejzik nije najpogodniji jezik za strukturalno programiranja obzirom da ne postoji mogućnost pozivanja potprograma uz prenošenja argumenata i definisanja lokalnih promenljivih (na primer, u dva



Najkompletniji stoni računar na tržištu: BBC B



Poslednji dani jednog uspešnog računara: Acorn Atom

različita potprograma može da se koristi promenljiva i a da računar definiše dva potpuno različita promenljiva) BBC bejzik, međutim, omogućava sve ovo i mnogo drugih stvari. Neke od njih su rezervisane za rad sa ekranom.

BBC model B ima maksimalnu rezoluciju 640 256. Međutim, „pamćenje" sadržaja ovakvog ekrana odnosi mnogo memorije (oko 20 Kb), pa bi korisniku ostalo premalo RAM-a. Zato su konstruktori pribegli kompromisu: uvedeno je osam modova. U modu nula, na primer, rezolucija je maksimalna (640 256), ali postoje samo dve boje, u modu jedan postoje četiri boje, rezolucija je 320 256, a ekran i dalje zauzima 20 Kb korisničke memorije. Već u modu pet rezolucija je samo 160 256 (četiri boje), ali ekran zauzima samo 10 Kb. U modu šest, najzad, može da se piše samo tekst uz utrošak minimalnog RAM-a (8 Kb). Za ljubitelje grafika visoke rezolucije je predviđen mod 2 u kome je na raspolaganju čak 16 boja, a za one koje slika ne interesuju — teletekst mod (mod sedam) u kome se na ekranu piše samo minimalna količina teksta uz minimalan utrošak RAM-a — svega 1 Kb.

Broj slova u redu takođe varira: u nekim modovima može da se raspolaže sa samo 20 slova (slova su tada, jasno, prilično široka), a u drugima čak i sa 80. Ovim su širom otvorena vrata primena BBC računara za obradu teksta. Kao posebna pogodnost ostavljena je mogućnost definisanja specijalnih karaktera (npr. grčka slova, matematički simboli, znaci na kartama i slično).

BBC računari su vrlo brzi — prema testovima brzina ni jedan drugi stoni računar iz ekonomske i srednje (a možda ni iz više) klase ne može da ih nadmaši kada se radi o brzini računanja i poređenja. To je postignuto originalnim i izvanrednim rešenjima u strukturi BASIC-interpretatora i naterivanjem procesora 6502 da radi na frekvenciji od 2 MHz. Brzina rada je takva da se mnogi programi za dinamične igre mogu pisati na bejziku, što unekoliko eliminiše potrebu za uvek neprijatnim mašinskim jezikom. I pored toga, u ROM je ugrađen takozvani editor/asembler — program koji omogućava jednostavno unošenja i ispravljanja mašinskih programa.

BBC B je opremljen časovnikom i veoma moćnim generatorom tonova: u istom trenutku mogu da se čuju čak tri tona, što otvara mogućnosti sinteze muzike ili, čak, glase. Naročito su značajni mnogobrojni ugrađeni interfejsi i zaista nesagledive mogućnosti ekspanzija. Pored uobičajenog priključka za crno-beli, odnosno kolor televizor, na zadnjoj strani su ugrađeni odvojeni priključci za crno beli i RGB kolor monitor na kojima se dobija daleko efektnija slika. U računar je ugrađen RS423 serijski interfejs, paralelni interfejs za štampač, interfejs za povezivanje sa disk-jedinicama, AD konvertor, Econet interfejs, a mogu se lako ugraditi Teletext prijemnik, ploča sa drugim procesorom (Z80A ili još jedan 6502) i drugo. Od skora je na raspolaganju i proširenja memorija koja je do sada predstavljala izvestan problem. U sam računar se priključuju moduli sa komercijalnim programima (npr. tekst-procesor, DOS i slično) koji mogu da se uključuju ili isključuju pod programskom kontrolom.

Acorn Electron

Najnoviji adut firme Acorn se zove Electron. Radi se o izvanrednom računaru koji košta svega 200 funti i koji će, bar prema našim procenama, ozbiljno uzdrmati pozicija mnogih drugih proizvođača stonih računara. Electron može da se okarakteriše kao BBC B u malom: računar je opremljen RAM-om od 32 Kb i istim tolikim ROM-om. U ROM je, kako izgleda, ugrađen program koji je praktično identičan sa onim BBC-jevim. Korišćeni bejzik je, dakle, praktično isti, čak i tamo gde su mogućnosti smanjena (generator tonova, na primer, ovde proizvodi samo jedan ton u jednom trenutku), naredbe nisu promenjene tako da se programi pisani za BBC računara mogu, praktično bez ikakvih prepravki, izvršavati na Electronu. Grafika i broj boja je identičan sa BBC-jevim, snimanje programa na kasetu se obavlja jednakom brzinom od 1200 bauda, pa se čak i programi pisani za BBC B mogu direktno učitati u memoriju Electrona i tamo izvršavati. Tastatura je profesionalna i ima jednu prednost u odnosu na BBC-evu: ispod svakog tastera je ispisana po jedna naredba pa korisnik može da bira da li želi da otkuca svaku reč slovo po slovo ili da pronađe taster kome je ona dodeljena. Electron nema odvojene tastere kojima mogu da se dodela proizvoljna funkcije (ovi tasteri predstavljaju jednu od važnih karakteristika BBC B), ali njihovu funkciju preuzimaju tasteri sa brojevima kada se pritisne CAPS LOCK.

Electron ima manje mogućnosti ekspanzije od svoga starijeg brata: na njegovoj zadnjoj ploči, pored priključaka za monitor odnosno televizor i napajanja, nema drugih izvoda. Za ekspanziju je predviđen port na koji tek treba da se dodaju potrebni

Sve prednosti BBC bejzika za manja od 200 funti Acorn Electron

interfejsi U trenutku kada ovo pišemo, Electron sa tek pojavio na tržištu pa hardverskih dodataka još nema, najviše sa oseća nedostatak štampača i disk-jedinica. Poznajući firmu Acorn, medutim, možemo da očekujemo da će sa ovi dodaci pojaviti pra početka 1984, kada će Elektron zauzeti msato koja mu po mogućnostima pripeda sam vrh tabela najtraženijih modela iz aradnje kiase na evropakom tržištu.

Da li će se, posla pojave Electrona, BBC A i B prodavati? BBC A vrlo verovatno neće ali amatramo da će BBC B biti čak i popularniji nego pre Raziog? Elactron je za 30% sporiji od njega, ima manje mogućnosti ekspanzije i teže sa povezuje u računarsku mrežu. Sa druge atrane, avi programi sa Elactrona (a jeftiniji računar će osvojiti daleko šire tržište od i dalje skupog BBC B) moći će da se primanjuju i na BBC B, pa će njegova programaka podrška, dobra i u ovom momantu, biti značajno unapređena Zato možemo samo da ponovimo zaključak Stive Mana (Steve Mann) iz časopisa „Personal Computer World" „Kada bih birso računar, svakako bih kupio BBC modei B ali ako ne bih imso 400 funti smatrao bih da je Electron fantastičan izbor!"

Računar pred kojim je budućnost· Lynx

Lynx

Za raziiku od Dragona 32, Lynx izgleda prilično skromno — očigiedno da na dizajn nije potrošeno mnogo vramena. Međutim, njegova karakteristika su prilično povoljne. Memorija ovog računara, u osnovnoj verziji, iznosi priatojnih 48 K, a može da sa proširi do 192 Ki Računar poseduje profesionalnu tastaturu, što ga čini pogodnim za obredu teksta U svakom redu u izvornoj verziji se prikazuju 40 karaktera, ali ovaj broj može da sa udvostruči dodavanjem posebnog modula Lynx je opremijen vrlo dobrim bejzikom i programi sa vrlo brzo izvršavaju. Iako je računar namenjan uglavnom za „ozbiljne" primane, rezolucija 248 256 omogućava realizaciju interesantnih igara

U momentu kada ovo pišemo, Lynx je nov na tržištu pa se može očekivati da njegova cene (255 funti) uskoro padne Programaka podrška ja zato prilično slaba, iako njegovi konstrukotri tvrde da su, za pisanja programa, potrebne samo minorne modifikacije rutina pisanih za drugé računara opramijene procesorom Z80A Sve u svamu, izgleda da Lynxu pripeda budućnost.

Commodore 64

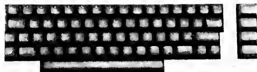
Firma Commodora je na tržište izbacila značajan broj dobrih stonih računara Prvi popularni modai je bio VIC 20 čija je osna nekoliko pute obarana — sada košta ispod 200 funti ali sa i dalje odlično prodaja

VIC 20, i pored niska cene, poseduje profesionalnu tastaturu, solidnu mamoriju (proširuje sa do 29 K), odličan generator tonova i velike mogućnosti akspanzije Grafika je, doduše, slaba ali može da se poveća dodavanjem posebnog modula Glavna mana računara je što u jednom redu prikazuje samo 22 slova, što ga čini neupotrebljivim za obradu tekata i mnoge poslovne primena Dodavanjem posebnih modula i ovaj broj može da sa poboljša, ali na treba zaboraviti da kupovina modula značajno povećava osnu računara

U Jugoslaviji je doste vlasnika VIC 20, ali neki od njih već pokušavaju da prodaju svoja kompjutere kako bi nabavili VIC 10 ili Commodora 64

VIC 10 (u Sjedinjanim Državama poznat kao MAC) je novi proizvod koji bi trabalo da ugrozi Spectruma Za to treba da sa pobrine niska cena od svega 100 funti i primemijive karakteristika nov video-čip koji omogućava grafiku visoke rezolucija u 16 boja, generator tonova i dobra tastatura. Niska cena je posledica činjenica da računar, u osnovnoj verziji, ima samo dva kilobajta memorije, što je beznačajno malo, pogotovu ako sa zna da računar „na zna" bejzik! Naima, posla svakog uključivanja računara traba sa kasete učitati bejzik interpratator i početi sa radom. Ovo učitavanje interpratatora ja osobina velikih kompjuterskih sistema· na taj način je računar nezavisan od programakog jezika i koranik, na primer, može da učita fortan umesto bejzika. No, za stone računare ova pogodnost se pretvara u svoju suprotnost· računar sa često uključuje i isključuje a učitavanja dugog sistemakog programa oduzima više minuta Kada znamo da većina korisnika želi da radi u bejziku, vidimo da se Commodora nija baš proslavio uvodeći ovakvu „fleksibilnost"

Mnogo veću popularnost ima šansu da stekna Commodora 64 (samo par meseci posla izlaska na tržište u Beogradu postoje bar tri ovakva računara!). Šifra 64 označava, naravno, 64 K memorija koja je na raspolaganju korisniku koji ja nabavio osnovni model koji košta samo 250 funti (ima indicija da će osna sići na 200 funti u vrlo skoroj budućnosti) Ovaj računar krase profesionalna tastatura, visoka rezolucija. odličan sat karaktera i velika mogućnosti proširenja Commodore je srazmerno brz računar sa relativno dobrim bejzikom (premda su neka naredba mogla da budu i bolje rešena) i, s obzirom da je na tržištu tek kratko, skromnom programakom podrškom Međutim, njegovi konstruktori su se postarali da omoguće korišćenja nekih programa za BBC računara na ovom kompjuteru, što bi trabalo da olakša zajednički razvoj softera Sve u svemu, čini nam sa da je kupovina Commodora 64 vrlo dobra odluka, teško da i jedan računar koji se danas nalazi na tržištu ima bolji odnos mogućnosti/cena. Commodore može odlično da posluži za igre (mada oni koji žele samo da sa igraju mogu jeftinije da prodju kupovinom Spectruma), ali i za vrlo složene naučnotehnička proračune i obradu teksta.



Najbolji odnos mogućnosti / cena Commodore 64

Proširan disk-jedinicama (sada se do kompletnog sistema koji obuhvata računar, disk-jedinice i štampač može doći za samo 600 funti), Commodora 64 postaje neobično pogodan za poslovne primene čak i kada je u pitanju obrada velike količine informacija.

Commodore proizvodi i neke skuplje računara većih mogućnosti (npr. VIC 30, 4016 i slično), ali nam se čini da njihova cena neće privući praviše Jugoslovena.

Dragon 32

Dragon 32 je računar vrlo lepog dizajna na kome dominira izvanredna profesionalna tastatura. Koristi mikroprocesor 6809E (ovaj procesor danas nije mnogo u upotrebi) i poseduje 32 K memorije u osnovnoj verziji (memorija može da se proširi do 64 K). Na ekranu generiše samo velika slova i ima grafiku visoke rezolucije 256 192 u devet boja. Opremljen je dobrim bejzikom i radi prilično brzo.

Pogled na ove karakteristika navodi na pomisao da je Dragon 32 namenjen „ozbiljnim" primenama. Međutim, praksa pokazuje da se ovaj računar koristi praktično isključivo za igre! To su najčešće igre-avanture tipa "Dungeons & Dragons" (odatle valjda i ime računara), ali ima i akcionih, dok su logičke igre veoma retke.



Ozbiljan računar koga shvataju neozbiljno Dragon 32

Programska podrška je više nego moćna: igre se prodaju na kasetama ili u posebnim modulima koji se priključuju na port računara; cena programa na kasetama je obično 5—6, a modula 15 funti. Ukoliko kupujete program za sebe, bolje je da nabavite modul obzirom da nećete morati da se brinete za probleme učitavanja sa kasete nastale usled nekompatebilnosti kasetofona. No, programe iz modula ne možete lako da presnimavate i dajete prijateljima.

Naš utisak o računaru Dragon 32 je prilično povoljan. Čini nam se da računar može da se koristi i u svim vrstama naučno-tehničkih proračuna, kao i u vežbanju programiranja. Za poslovne primene nije dobar obzirom na nedostatak disk-jedinica i mali broj interfejsa.

TRS 80

Opisom TRS 80 dolazimo do računara iz srednje klase. Čuvena firma Tandy (u Sjedinjenim Državama prodaje uređaje pod imenom Radio Sheck, otuda skraćenica TRS) predstavlja verovatno najpoznatije ime kada je reč o mini-računarima, a tu reputaciju treba, pre svega, da zahvali kompjuterima iz klase TRS 80.

Sve je počelo pre pet-šest godina kada se na tržištu pojavio TRS 80 model I koji i dalje uživa priličnu popularnost među vlasnicima računara. Ovaj računar, je, na neki način, izrađen „klasično": njegov dizajn je inspirisao mnoge docnije modele. Računar, sa današnjeg aspekta, ima mnogobrojne mane: rezolucija je zaprepašćujuće mala (128 48 tačaka), nije opremljen generatorom tonova, radi sa crno-belim televizorom, nema ugrađenih interfejsa... Međutim, TRS 80 I je bio najbolje što se u trenutku kada je konstruisan moglo napraviti. Pri njegovoj izradi nije se škrtarilo: hardver je dobro i profesionalno urađen (iako bi mnoge stvari mogle da budu jednostavnije), bejzik je dobro koncipiran i, što je najvažnije, spreman za dodavanje novih naredbi. Naime, većina drugih računara je opremljena određenim brojem naredbi i korisnicima je ostavljeno da svoje mašinske potprograme (koji, u suštini, obično dodaju neke nove mogućnosti) pozivaju naredbama CALL odnosno USR. Kod TRS 80 I nije tako: ROM je opremljen određenim brojem segmenata koji izvršavaju neke delove RAM-a. U trenutku kada uključimo računar, ti delovi RAM-a se pune mašinskim naredbama RTN (RETURN=vrati se) i njihovo pozivanje ne izaziva nikakve posledice. No korisnik može da te RTN naredbe zameni pozivima sopstvenih potprograma i tako „proizvede" naredbe koje su mu potrebne.

TRS 80 I je obično opremljen takozvanim level II bejzikom. Ova varijante bejzika je vrlo fleksibilna i omogućava jednostavno ostvarivanje čak i vrlo složenih sekvenci. Ukoliko ovaj bejzik nije dovoljan, može se učitati dodatak nazvan level III bejzik: time se dodaju nove naredbe za rad sa ekranom, prenumerisavanja programa, kontrolu časovnika.

TRS 80 I ima značajne mogućnosti ekspanzija, ali je ta ekspanzija prilično skupa. Potrebno je, najpre, nabaviti takozvani Expansion Interfejs. To je uređaj veličine samog računara u kome se nalazi disk-kontroler, nekoliko priključaka za razne periferijske uređaje i sklop za kontrolu dva kasetofona. Tek uz pomoć Expansion Interface-a na TRS 80 I mogu da se priključe mnogobrojne disk-jedinice i štampači, što proširuje ovaj kompjuter do moćnog sistema.

TRS 80 I je uglavnom namenjen „ozbiljnim" primenama. Za njih je ovaj računar vrlo dobro opremljen ne samo hardverom nego i programima. Kompjuter, naime, ima moćnu programsku podršku, najmoćniju posle čuvenog računara Apple II. Ti programi su često prilično skupi, prvenstveno zato što su namenjeni ljudima koji će korišćenjem računara, smanjivati materijalne troškove svoga poslovanja. Postoji, jasno, doste programa za igre, ali izgleda da one nisu bile popularne kao danas u doba kada se TRS 80 I pojavio na tržištu. Programi za igre na ovom računaru su, u poređenju sa programima za ZX računare i Dragon 32, pravi amaterski počeci.

Sumirajući sve navedeno, nismo sigurni da kupovina TRS 80 I danas predstavlja dobru investiciju. Pre svega, računar nije jeftin (engleska cena 290 funti), a dodaci su još skuplji. Osim toga, TRS 80 I se danas malo (ili ni malo) proizvodi — nasledio ga je model III.

Na prvi pogled izgleda besmisleno da model III nasleđuje model I. Šta je sa modelom II? TRS 80 II postoji, ali je to računar sasvim druge koncepcije — kompjuter namenjen poslovnim primenama koji košta par hiljada funti i, kao takav, nije naročito povoljna prilika za naš džep. Model III, međutim, košta „samo" 1500 funti, ali je opremljen sopstvenim monitorom i, po želji, jednom ili dvema disk-jedinicama. U samom računaru se nalazi disk-kontroler i interfejs za povezivanje sa štampačima i drugim periferijskim uređajima. TRS 80 III, dakle, predstavlja zaokruženu celinu. Opremljen je Level III bejzikom i memorijom od 16 kilobajta koja može da se proširi do 48 K i profesionalnom tastaturom. Većina programa pisanih za model I može bez problema da radi na modelu III; neki mašinski programi, doduše, zahtevaju određene prepravke, ali je to posao koji može da obavi i osrednje upućeni programer. Kao i model I, model III je uglavnom namenjen „ozbiljnim" primenama, mada može da se dođe i do određenog broja programa za igre. U Beogradu su TRS 80 računari prilično popularni i sa te strane nije loše nabaviti neki od ovih modela — može se lako doći do odličnih programa. Model III, mora se reći, pri kupovini zahteva rešavanje još jednog problema: čitav uređaj je veliki (zahvaljujući ekranu) i „vrlo ozbiljno" izgle-



Za ljubitelje računara skromnijih potreba TRS 80 MC 10

da što može da donese problema sa cari-nom No, ako se ovaj problem i problem cena nekako reši, kupovina TRS 80 III predstavlja dobru investiciju

Tandy proizvodi i druge računare koji nose ime TRS 80 Tu je, na primer, TRS 80 color, računar skromnih mogućnosti namenjan korisnicima sa plićim džepom Ovaj računar košta nekih 200 funti u verziji sa 4 K memorije Na njemu se ne mogu koristiti programi pisani za „veće" Tandy računare jer umesto Z80A procesora koristi 6809E Računar je opremljen skromnim bejzikom koji može da se proširi dodavanjem posebnog modula RAM može da se poveća do 32 K ali takvo proširenje košta gotovo 200 funti Tastatura je poluprofesionalna, a rezolucija dobra 256 192 TRS 80 color je namenjen prvenstveno onima koji žele da se igraju, ali izgleda da tu nije postigao naročitu popularnost iako je na tržištu već više od godinu dana, programska podrška mu je prilično skromna.

TRS 80 model 100 je novi proizvod koji se nalazi na prelazu između džepnih i stonih računara Ovaj model će verovatno steći značajnu popularnost, a njegova karakteristike su već opisane u „Galaksiji"

Na kraju, nedavno se pojavio TRS 80 model 16 Reč je o moćnom računaru koji bi firmu Tandy trebalo da postavi na jedno od najviših mesta u svetu kućnih kompjutera Prateći nova trendova, računar je opremljen šesnaestobitnim procesorom, što je omogućilo veliku memoriju i brz rad Firma Tandy se, međutim, pobrinula za svoja „stara mušterije" računar poseduje i procesor Z80A koji omogućava izvršavanja programa pisanih za ranije modela Ipak, TRS 80 model 16 je previše skup da bi zaslužio da ga preporučimo ljubiteljima kućnih računara

Newbrain

Slična sudbina, prema našim ocenama potpuno nezasluženo, zadesila je i Newbrain Newbrain je, kako nam se čini, dobar računar Opremljen je sopstvenim displejom (verzija AD) i može da se proširi pakovanjem baterija i tako postane neka vrsta prenosivog stonog računara Priključivanjem na standardni televizor, Newbrain omogućava grafiku visoka rezolucije 640 256 Njegova tastatura se sastoji od dirki koje se kreću kada ih korisnik pritiska, iako ne postoji potpuni osećaj koji pruža profesionalna tastatura Memorija je gotovo neverovatna u osnovnoj verziji 32 Kb, a u maksimalnoj — 2 Mb (1 Mb = 1024 Kb)! Dva megabajta je ogromna memorija koju mogu samo da požele čak i veliki kompjuterski sistemi i neusklađena je sa drugim mogućnostima računara — snimanje čitavog njenog sadržaja na kasetu bi, na primer, moglo da potraje dobra tri časa, a računar nije sposoban za povezivanje sa disk-jedinicom (posebno je pitanje što dva megabajta ne bi mogla da stanu na jednu disketu, morao bi da se koristi hard-disc koji, čak i u sadašnjim uslovima, ne spada ni u opremu stonih računara najviše klasa) Zato će Newbrain verovatno biti korišćen sa svoja 32 Kb

U ROM računara je ugrađena i jedna novost — bejzik kompajler On omogućava prevođenje bejzik programa na mašinski jezik i, samim tim, njihovo vrlo brzo izvršavanje Svaki bolji programer veoma ceni ovakvu karakteristiku, ali za početnika ona ništa ne znači, njemu pada u oči da kompjuter nije opremljen generatorom tonova, da ne daje sliku u boji, da na tržištu nema dovoljno programa pisanih za njega Kombinovane sa neatraktivnim dizajnom Newbraina, ove karakteristike su učinile da se premalo ljudi odluči za njega što, u uslovima nemilosrdnog tržišta, vodi pravo u bankrotstvo To se dogodilo proizvođačima ovog računara i to samo nekoliko dana pre zaključivanja ovog specijalnog izdanja Newbrain je prodat jednoj holandskoj firmi koja treba da pripazi da je ne snađe slična sudbina Iako je, po našoj oceni, Newbrain dobar računar (ne isplati se jedino kupovati pakovanja baterija koja košta 80 funti i modela AD), ni u kom slučaju ne možemo da vam savetujemo njegovu nabavku u sadašnjim uslovima pitanje je da li će servis biti obezbeđen, programska podrške i časopisa neće biti, broj vlasnika računara se neće povećavati da bi se pojavili hardverski dodaci Ako vam je potreban prenosiv računar značajnih mogućnosti, savetovali bismo vam TRS 80 model 100, HP75C ili CC40.

Jupiter Ace

Jupiter Ace se nalazi na prelazu između ZX81 i Spectruma Za razliku od najvećeg broja stonih računara, Jupiter Ace ne poznaje bejzik — njegov „maternji jezik" je forth O forth u ćemo više govoriti u poglavlju koje je posvećeno komercijalnim programima, ali već sada moramo da kažemo da je reč o jeziku nižeg nivoa koji se nešto teža ući, ali zato omogućava znatno brže izvršavanje programa i fleksibilniju upotrebu memorije Jupiter Ace ima ROM od 8 K (u koji je, s obzirom na nivo jezika, stalo dosta naredbi) i RAM od 3 kilobajta To, jasno, nije baš mnogo, ali ne treba zaboraviti da se ova memorija troši daleko racionalnije nego kod ZX81 Za one kojima je 3 Kb premalo, na raspolaganju je modul od 16 Kb koji košta tridesetak funti Tastatura je po karakteristikama približna Spectrumovoj, iako nešto jednostavnija za upotrebu Računar na ekranu prikazuje velika i mala slova, kao i slika koristeći grafiku niska rezolucije Časovnik i biper dopunjuju



Nezaslužen krah na tržištu Newbrain

„arsenal" ovog kompjutera koji je namenjen igrama

Jupiter Ace se na tržištu nalazio samo godinu dana, a onda je nastupio neizbežno bankrotstvo njegovih konstruktora Njega su dobri poznavaoci kompjuterskog tržišta mogli da predvide od samog početka ugrađivanje novog programskog jezika (fort) u ROM računara je procenjeno kao neka vrsta „promašena tema" u ovom trenutku svi pomalo gunđamo na neka karakteristike bejzika (ponajviše na to što je spor jezik) ali niko ne može da negira njegovu apsolutnu dominaciju na tržištu Računar koji radi na jeziku koji je bliži mašinskom i samim tim relativno složen za širok krug korisnika (korisnici kojima je ovaj jezik blizak su iskusniji programeri koji će se teško odlučiti na kupovinu jeftinog kompjutera kakav je Jupiter Ace) i koji je, uz to, vrlo malo programski podržan (u časopisima treba „svećom tražiti" programe za njega) svojim konstruktorima i ne može da donese ništa drugo osim bankrotstva iako će u nekim engleskim radnjama Jupiter Ace moći da se nabavi još izvesno vreme, čini nam se da je kupovina ovog računara u sadašnjim uslovima veoma loša odluka

Hewlett-Packard

Hewlett-Packard je poznata američka firma koja se bavi proizvodnjom mernih uređaja, sistema za kontrolu procesa i računara U proizvodnji stonih računara Hewlett-Packard je jedan od pionira njihovi modeli 9825 i 9845 predstavljaju sisteme koji su u trenutku pojava na tržištu izazvali pravu revoluciju, ali koji su i danas neobično popularni i, često, neprevaziđeni

Može da se kaže da koncepcija Hewlett-Packardovih računara predstavlja suštu suprotnost koncepciji Sinklerovih Firmu Hewlett-Packard jedino interesuju mogućnosti i kvalitet izrada — cena je sasvim nebitna Zato su njihovi računari neobično



Igrao je samo jedno leto Jupiter Ace

dobri, ali i neobično skupi — često prikladni samo za američki džep Ako se odlučite na kupovinu Hewlatt-Packardovog računara, možete da budete sigurni da ćete doći do uređaja koji je neobično dobro urađen, koji se retko ili nikada ne kvari, možete da računate da ni jedno pismo koje napišete firmi neće ostati bez odgovora stručnog lica, da ćete imati na raspolaganju širok izbor jeftinih programa (ali ne i preširok izbor programa za igre — Hewlett-Packard interesuju uglavnom ozbiljne primene računara) i obilje stručnih časopisa koje pišu najstručniji profesionalci

U ovom kratkom prikazu nećemo govoriti o ranim Hewlatt-Packardovim računarima, iako to predstavlja pravu nepravdu prema istoriji razvoja računara Počećemo od modela 85 koji, iako star pet godina, i danas nalazi veliki broj kupaca Cena ovog računara je čitavih 1960 funti; za taj novac se dobija memorija od 16 K (može da se proširi do 120 K a i dalje), ROM od 32 kilobajta u koji je smešten vrlo fleksibilan BASIC, profesionalna tastatura, ugrađen monitor sa 16 redova (32 slova u redu) i rezolucijom 192 256 U računar je ugrađen izvanredni specijalizovani kasetofon koji omogućava direktnu poslovnu primenu Ukoliko je potrebna obrada veće količine informacija, može da se doda disk jedinica ili hard-disk kapaciteta oko 2 Mb (obzirom da diskove proizvodi Hewlatt-Packard, može da se zamisli „da su i njihove cene „pristojne") U celini gledano, HP85 nije pogodan za igre a i programska podrška u toj oblasti nije mu jaka strana.

HP87 je namenjen onima kojima je potreban računar ogromnih mogućnosti za naučno-tehničke i poslovne primene i koji su spremni da za njega plate 2500 funti Računar poseduje memoriju od 128 K koja može da se proširi do 640 K ROM ima 48 kilobajta (!) i sadrži bejzik interpretator velikih mogućnosti HP 87 je, naime, koncipiran tako da omogući jednostavno pisanje programa koji će moći da izvrše vrlo složene zadatke Zato se u ROM-u nalaze programi koji podržavaju mnogobrojne naredbe koje drugi računari ne poznaju A stara je istina da veći broj naredbi u ROM-u čini računar bržim, jednostavnijim za upotrebu i — skupljim

Kao i HP85, HP 87 poseduje sopstveni crno-beli monitor, ali je ovaj znatno veći — u jednom redu staju 64 slova Rezolucija je pristojna, ali verovatno nedovoljna za neke igre, u računar je ugrađen časovnik sa kalendarom i generator tonova koji je pre namenjen upozoravanju korisnika da je neki posao obavljen nego sintezi muzike i glasa Računar, najzad, poseduje mogućnost povezivanja sa brojnim uređajima pomoću HP-IB interfejse

HP86 je, navodno, jeftin računar namenjen širem krugu korisnika Cena ovog „jeftinog" računara, u osnovnoj opremi, iznosi „samo" 1300 funti Za taj novac se dobija računar se 16 K memorije (može da se proširuje) koji se priključuje na neki crno-beli monitor Druge karakteristike HP86 su prilično dobre, ali ovaj računar ipak nije naročito povoljna investicija ukoliko tražimo jeftiniji računar, ne treba da razgledamo Hewlatt-Packardove kataloge!



Računar koji je odoleo zubu vremena Apple II

Apple II

Apple II je relativno star i apsolutno slavan kompjuter Praktično da nema ljubitelja elektronike i računara koji bar nekada nije ugledao tradicionalnu jabuku (znak firme Apple) i čuo reči pohvale od nekog vlasnika Apple

Sve je počelo prilično skromno dva mlada svršena studenta elektronike su, maltene u garaži, sastavili računar i nazvali ga Apple I Proizveden je samo mali broj primeraka koji su se dobro prodali i doneli dovoljno novca da se deo elektronike predizajnire, napiše novi ROM i — nastane Apple II Ovaj računar je izazvao pravi bum na tržištu kupilo ga je više stotina hiljada amatera i gotovo milion fabrika i preduzeća (do sada je prodato oko 1 700 000 primeraka računara Apple II, što je rekord koji čak ni Sinclairovi modeli neće nikada dostići) U to doba (1978 godina), na tržištu nije bilo mnogo računara prilagođenih širokom krugu korisnika, pa su programi pisani uglavnom za Apple II i TRS 80 Zato Apple II ima moćnu programsku podršku — moćniju nego i jedan drugi računar

Karakteristika Apple su uglavnom povoljne memorija od 48 K može da se proširi do 64 K, što je sasvim dovoljno za mnoge primene U ROM od samo 12 K stao je bejzik i posebni mašinski program za praćenje stanja sistema i programiranja na mašinskom jeziku (ovaj program se obično naziva program-monitor) Iako je bejzik ostvaren pomoću manjeg programa, prilično je fleksibilan i, razume se različit od bejzika drugih računara Teško je reći da li je Applov bejzik bolji ili gori od standardnog Microsoftovog — ima dosta argumenata u prilog svakoj od ove dve teze Bilo kako bilo, Applov bejzik je sasvim dovoljan za sve primene

Pored bejzika, Apple II može da se opremi paskalom, fortranom, kobolom, lispom, logom i drugim programskim jezicima Na zadnjoj strani računara se nalaze portovi u koje se dodaju specijalne „kartice" dodatak jedna od njih, na primer povećava broj karaktera u redu od 40 na 80, druga dodaje grafiku visoke rezolucije u boji, treća omogućava korišćenje programa pisanih za TRS 80 i druge računara koji koriste Z80 procesor i slično Osnovna verzija računara na ekranu prikazuje samo velika slova dok mala mogu da se dobiju dodavanjem posebne kartice Računar se povezuje sa standardnim kasetofonom i ta veza mu je slaba tačka — postiže se brzina snimanja od svega 300 bauda, što znači da se 48 K memorije snimi na traku za „samo" 20 minuta Problemi se upisivanjem na kasetu se rešavaju korišćenjem disk-jedinica (uračunate u cenu od 700 funti) — na svaku disketu staje po 140 kilobajta

Pronalazači Apple su uradili jednu malo neobičnu stvar koja se pokazala pametnom nisu svoj računar zaštitili patentima To je navelo mnoge nezavisne firme (ne računajući firme u Tajvanu i Hong-Kongu koje se ionako mnogo ne obaziru na patente) da proizvode modifikacije i hardverske dodatke za Apple Ti hardverski dodaci nisu mnogo potrebni (Apple II i sam ima dovoljno mogućnosti), ali predstavljaju dobar način da se korisniku izbaci više novca iz džepa U svetu su nastali mnogi klubovi korisnika računara Apple II, što ga je dalje popularizovalo i poboljšavalo prodaju.

Apple II je kod nas vrlo popularan, pa se čak i sklapa u Jugoslaviji Ne bismo, ipak mogli da vam preporučimo kupovinu nekog „domaćeg Apple" — cena od sedamdesetak starih miliona je, ipak, „pomalo" pretarana

Atari

Atari je firma koja se proslavila video-igrama, dok je u proizvodnji stonih računara relativno „nova“ Atari pokušava da stekne popularnost pomoću računara Atari 400 (košta 200 funti) i Atari 800 (400 funti) Bilo koji od ova dva modela da kupite, nećete moći da ga upotrabite bez posebnih modula Ovi računari, naime, ne poseduju ni jedan programski jezik U posebne portove priključuju se moduli sa raznim programima: popularni su moduli sa Atari TV igrama koja su postala pristupačna onima koji žele da nabave računar opšte namene ali je modul sa bejzikom, svakako, daleko potrebniji Oba računara, u osnovnoj verziji, imaju RAM od 16 K Atari 800 može da se proširi do 48 K (proširenja košta 130 funti), a Atari 400 ne može! Naima, nema nikakvih problema da se i model 400 proširi do iste konfiguracije, ali firma Atari to ne želi da čini da ne bi ugrozila prodaju skupljeg modela Mnogi prodavci, jasno, nemaju ništa protiv da prošire Atari 400 do 32 ili 48 K, ali upozoravaju kupce da je na taj način garancija postala nevažeća! Čini nam se da je ovo veoma neposlovna politika ove poznate firme

Atari 400, inače, ima senzorsku tastaturu koja nije naročito pogodna za kucanje programa Iako model 800 ima profesionalnu tastaturu, njegov bejzik, očigledno, nije pravljen tako da stimuliše korisnika da piše sopstvene programe Zato Atari računari ostaju dobri jedino za one koji žele da se igraju izvanrednim TV igrama primenjanim pomoću kompjutera što im, čini nam se, ne ide mnogo u prilog

IBM PC

Svi znamo šta u svetu računara znači ime IBM još od početka ovog veka i njenog pionira, čuvenog američkog naučnika i biznismana Holerita, IBM je neprevaziđena firma u proizvodnji velikih kompjuterskih sistema. IBM ulaže gigantska sredstva u razvoj novih računara, pa je normalno što okuplja najbolje kompjuterske stručnjake koje svet ima IBM do skora nije bio zainteresovan za mini-računare — ostvarivao je dovoljne profite prodajom velikih kompjutera Nije baš jasno zašto je ova firma pre dve godine odlučila da proizvede stoni računar — jedni kažu da je želela jednostavno da pokaže da i ovde može da stekne primat a drugi da je bilo neophodno zaposliti tim mladih stručnjaka koji treba da se afirmišu pre nego što steknu pravo da rade na većim projektima. Bilo kako bilo, IBM je proizveo računar koji danas, kako saznajemo, vlada na američkom tržištu

Američko tržište se po mnogo čemu razlikuje od evropskog — izgleda da na njemu slabo prolaze jeftini računari Za iole dobro stojećeg Amerikanca 2500 dolara ne predstavlja veliki izdatak i za njega je od interesa da nabavi dobar uređaj Baš zato hiljade Amerikanaca danas poseduje IBM PC Ovaj računar u osnovnoj verziji ima čitava 64 kilobajta memorije i može da se proširi do 256 K ROM od 40 K sadrži izvanredan Microsoftov bejzik i program za rad sa mašinskim jezikom — računar koristi u poslednje vreme neobično popularni šesnaestobitski procesor 8088

IBM PC je neobično pogodan za obradu teksta i poslovne primene pošto u osnovnoj verziji poseduje dve disk-jedinice (na disketu staje 160 K), a može da se priključi hard-disk na koji staje 10 Mb podataka. Grafika i ostale karakteristike ga čine pogodnim i za igre, ali je programska podrška na ovoj strani prilično slaba Numerička preciznost je dovoljna za većinu naučno-tehničkih primena za koje je posebno značajan i brz rad ovog računara Jedna od suštinskih tačaka koncepcije IBM PC je bila da se obezbedi veliki set karaktera i dosta tastera kojima mogu da se dodaljuju funkcije — u tome se u potpunosti uspelo!

IBM PC, dakle, predstavlja dobar izbor, ali za one korisnike koji imaju dubok džep i planiraju da koriste računar u „ozbiljnim“ oblastima Za početnika i one koji žele da se igraju IBM PC je, zaista, veliki luksuz.

Mašina od koje svi strepe IBM PC *mlađi*

IBM PC mlađi

Prvog novembra 1983 firma IBM je zvanično objavila karakteristike novoga modela koji će, po svoj prilici, osvojiti još šire tržište od IBM PC — novi računar se zove PCjr (PC mlađi) i poznatiji je po nadimku „kikiriki“ PCjr se prodaje u dve verzije — cenovna verzija obuhvata centralnu jedinicu sa 64 Kb ROM-a i 64 Kb RAM-a i košta 669 dolara, a proširena verzija, za relativno skromnih (za džep Amerikanca) 1269 dolara, uz to, obuhvata i dve disk-jedinice 128 Kb RAM-a, dodatak za poboljšanu grafičku rezoluciju, nekoliko modula i druge sitnice

PCjr donosi nekoliko novitata tastatura je potpuno odvojena od centralne jedinice računara i se njom komunicira infra-crvenim zracima ili, ako u prostoriji ima nekoliko tastatura, posredstvom kabla U ROM računara je ugrađen bejzik, editor-asembler (program za rad sa mašinskim jezikom procesora 8088) i par demonstracionih programa Jedan od njih, „Keyboard Adventure“, pisan je tako da zainteresuje svakog korisnika za IBM PCjr Trošenje ROM-a na program za igru je nešto novo u svetu stonih računara, ali je prilično uobičajeno kod većih kompjuterskih sistema (kakve IBM uglavnom i proizvodi).

U okviru informacije koja nam je jedino bila na raspolaganju prilikom pripreme ovog priloga nismo mogli da nađemo dovoljno podataka o onim karakteristikama novog kompjutera koje obično pominjemo Saznali smo, na primer, da se u svakom redu ekrana prikazuje po 40 slova i da ovaj broj, dodavanjem posebnog modula ili kupovinom proširene verzije, može da se poveća na 87, ali ne i sa koliko tačaka u svakom redu korisnik raspolaže i koliko boja može da koristi Obzirom da se računar priključuje na standardni kolor televizor ili, prema preporuci firme, na IBM kolor monitor, možemo da pretpostavimo da je rezolucija dobra Generator tonova je, naravno, uključen u obe verzije računara, a na samom kompjuteru postoje priključci za povezivanje sa štampačem, optičkim perom, palicama za igre, modemom i kasetofonom koji će biti posebno interesantan za one koji ne kupuju disk-jedinice

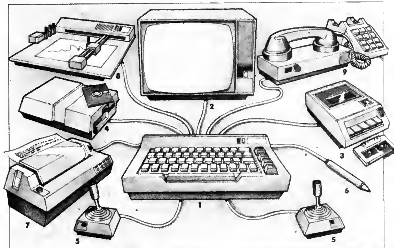
O bejziku koji novi računar koristi nemamo mnogo podataka ali, obzirom da je za ROM odvojeno 64 kilobajta adresnog prostora, možemo da pretpostavimo da je izvanredan Što je najvažnije, obezbeđena je delimična kompatibilnost sa IBM PC, pa će novi računar na startu biti opremljen izvesnom količinom izvanrednih programa

Računar je propraćen dobrom dokumentacijom koja obuhvata uputstvo za upotrebu, bejzik za početnike i abstrakt svega toga za iskusnije programere ili one koji su već radili sa IBM PC IBM pretpostavlja da će širok krug mladih ljubitelja računara nabaviti njihov kompjuter, upoznati i zavoleti njegove karakteristike i u budućnosti se opredeljivati za moćnije (i skuplje) modele iste firme Verovatno je zato cena PCjr, za ono što nudi, relativno niska firma želi da investira u svoje buduće kadrove Vrlo je verovatno da će novi kompjuter postići ogromnu popularnost na američkom tržištu, a možda i na evropskom (cene u Evropi još nisu objavljene, ali se, naravno, može očekivati da će biti više od američkih) Jugoslovenima će, obzirom na kurs dolara, novi IBM-ov kompjuter najverovatnije biti predstavljen samo na slikama poput ove

Periferijska oprema

3/ vrata u svet

Računer je, na prvi pogled, neobično ekonomične napreve: jednom kupljen, on ne troši praktično ništa, a može da obavlja neobično složena poslove i posluži kao nepresušan izbor zabave. Na žalost, nije baš tako. Mnogi ljubitelji računera, posle nakoliko mesaci zabave, posteju zaraženi nečim što se u žargonu nazive „hekarska klica" (izraz nije pogrdan — označeve samo zegriženog ljubitelja računara) i požele da povečaju mogućnosti svog „elektronskog ljubimca". Mogućnosti se povečavaju kupovinom periferijske opreme koja je, po pravilu skuplja od samog računara. Pri tom večine korisnike dolazi u situeciju da višestruko plati ono što je uštedela prilikom nabavke računara.



Porodica na okupu Kućni računar (1) praktično ne može da se koristi bez televizora (2) i kasetofona (3) za smeštaj programa i podataka, disk-jedinica (4) ima znatno veći kapacitet i kraće vreme pristupa podacima ali je i neuporedivo skuplja, palice za igru (5) i optička olovka (6) spadaju u najefikasniju periferijsku opremu, štampač (7) je potreban svakome, a ploter (8) za izradu crteža i modam (9) za uključivanje kućnog računara u centralnu računarsku mrežu samo profesionalcima

KOME ŠTA TREBA+)

Primena računara	štampač	disk-jed.	ploter	modem	palice
Igre	10	5	2	20	80
Naučno-tehnički programi	60	30	40	10	2
Poslovne primene (manji poslovi)	95	60	5	35	2
Poslovne primene (veliki posl.)	100	95	5	50	2
Programiranje (vežba)	30	10	2	10	2
Programiranje („ozbiljno")	70	30	5	50	2

+) Brojevi označavaju neku vrstu rejtinga. Veći broj označava da je pojedini periferijski uređaj potrebniji za neku primenu računara.

štampači

Pod periferijskom opremom podrazumevamo sva uređaja koji se povezuju sa računarom. U periferijsku opremu, po definiciji, ne spadaju razni RAM i ROM moduli koji se montiraju u sam računar, niti promene nekih njegovih delova (nabavka bolje tastature, na primer). Na tržištu se nalazi pravo mnoštvo periferijskih uređaja: neke od njih prave proizvođači računara i namenjuju striktno njihovim modelima, a druge nezavisne firme koje opremu prodaju u nekoliko verzija — za razne računare. Neki periferijski uređaji su prilično skupi i po ceni mnogostruko prevazilaze računar za koji su nabavljeni, dok drugi ne koštaju mnogo. Kupovina svakog od njih, ipak, neumitno izvlači novac iz džepa ljubitelja računara koji tako pada u svojevrsnu zamku: verovatno bi se retko ko odlučio da kupi računar sa mnoštvom periferijske opreme kada bi video ukupnu cenu. Rastegnuto u par godina, ovakva nabavka zahteva česte manje izdatke, ali ukupna suma postaje značajna. Zato je veoma značajno odabrati one periferijske uređaje koji su zaista potrebni i koji će, kroz upotrebu, vratiti novac koji je u njih uložen ili priuštiti korisniku odgovarajuću dozu zabave.

Namena

Pri procenjivanju potreba za nabavku nekog uređaja treba biti veoma rigorozan. Da li razmišljamo o kupovini nekog uređaja zato što nam je on potreban ili zato što bismo želeli da vidimo kako radi? Da li će on, kada ga kupimo, biti praktično upotrebljiv na našem računaru (ako imamo računar male memorije sa lošom tastaturom, ima li smisla opremiti ga profesionalnim i skupim štampačem)? Najzad, da li smo, među raspoloživim periferijskim uređajima, izabrali onaj čije nam karakteristike zaista odgovaraju? Nema nikakvog smisla kupovati loš štampač da bismo imali štampač, ili jednostruki flopi-disk koji zahteva držanje operativnog sistema na svakoj disketi. Ukoliko ste se odlučili da nabavite neki uređaj, sačekajte da sakupite dovoljno novca da kupite neki sa prihvatljivim karakteristikama. Sa druge strane, ne treba kupovati preskupe uređaje čije su osobine dobre ali za nas nepotrebne.

Kompatibilnost

Ukoliko kupujete periferijsku opremu koju je proizvela firma čiji ste računar kupili, ne treba mnogo da brinete. Uređaj će se lako priključiti na vaš računar i dobićete uputstvo za upotrebu koje će detaljno objasniti naredbe koje upotrebljavate da biste pokrenuli uređaj. No, ako se odlučite za kupovinu dodatka koje proizvodi neka nezavisna firma, budite vrlo oprezni. Moguće je, pre svega, da se uređaj ne može priključiti na vaš računar ili da to priključivanje zahteva dogradnju posebnog interfejsa. Ukoliko u reklami vidite da se uređaj pravi specijalno za vaš računar, možete da očekujete da ćete dobro proći sa povezivanjem. Ali, ako se u reklami pominje pet-šest vrsta računara, sasvim je moguće da ćete, uz kupljeni uređaj, dobiti uputstvo u kome piše „za povezivanje sa računarom potreban je interfejs koji naša firma planira vrlo skoro da proizvede. Pratite naše oglase".

Provera mogućnosti povezivanja je najsigurnija na licu mesta. Zahtevajte od prodavca da vam prikaže kako uređaj radi povezan sa računarom koji vi imate i obratite pažnju na ono što vam pokazuje. Ako umesto vas kupovinu obavlja neko drugi, nije loše da se prethodno raspitate kod nekog ko već ima takav uređaj ili da posetite neki kompjuterski klub.

Kod kupovine periferijskih uređaja nezavisnih proizvođača pojavljuje se još jedan problem: može da se desi da kupite uređaj koji se bez problema povezuje sa vašim računarom, ali da ostanete bez programa za njegovu kontrolu. Dobar primer toga su disk-jedinice za BBC računare: one koje prodaje Acorn lepo rade ali se na tržištu mogu naći i jeftinije. No, ako kupite disk-jedinicu nekog drugog proizvođača, ostaćete bez programa za formatiranje diskete bez koga je čitav uređaj beskoristan. Neki prodavci će vam, doduše, dozvoliti da prekopirate Acornovu sistemsku disketu da bi vas nagovorili da kupite uređaj baš kod njih — u svakom slučaju, treba otvoriti „četvore oči".

Kao i računari, i periferijska oprema je podeljena na klase. Ekonomska klasa je ovde malobrojna i svodi se na periferijske uređaje za Sinklerove modele. Srednja i više klase su jasno razdvojene: prva je namenjena onima koji upotrebljavaju računar za vođenje manjih poslova i naučno-tehnička primene, a druga za složene poslovne primene.

Primerenost

Periferijska oprema treba da bude prilagođena računaru. Ako nabavimo jeftin računar želeći da se igramo sa njim, nema mnogo smisla da ga opremamo skupim dodacima jer on neće moći da ih kontroliše na zadovoljavajući način. Možemo, takođe, da kupimo skup računar i da docnije uvidimo da nam je potreban printer koji ćemo koristiti u manjoj meri — samo za ispisivanje programa i slično. Tada možemo da se opredelimo za nabavku štampača iz niže ili srednje klase koji će nam u praksi biti dovoljan.

Tehnički kvalitet

Kod periferijske opreme tehnički kvalitet izrade je neobično važan. Računar je uglavnom sastavljen od elektronskih komponenti koje se retko kvare (pod pretpostavkom da nisu izrađene sa greškom). Štampači i disk-jedinice, naprotiv, sadrže vrlo precizne mehaničke delove koji su kontrolisani elektronikom. Česta upotreba može da dovede do manjih oštećenja pojedinih delova koja, posle izvesnog vremena, mogu da onemoguće ispravan rad uređaja. Zato je za one koji planiraju poslovnu primenu računara neophodno da nabave kvalitetno izrađene uređaje: oni će, možda, u startu biti znatno skuplji, ali će se uštedeti na izdacima za popravke. Ne treba da zaboravimo da je kod nas popravka nekog ovakvog uređaja ili slanje u strani servis skopčano sa mnogim problemima.

Najčešće korišćeni periferijski uređaji su štampači, disk-jedinice, ploteri, telefonski modemi i palice za igre. Pre nego što pređemo na njihove karakteristike, dajemo tabelu koja prikazuje koji korisnici, s obzirom na primenu računara koju planiraju, imaju potrebu za nabavkom pojedinih periferijskih uređaja.

Štampač je, po našoj oceni, najpotrebniji periferijski uređaj. Bez njega, računar nema nikakve mogućnosti da ono što je uradio prezentira na način koji je pogodan za umnožavanje i korišćenje u širem krugu ljudi. Štampač, pored ostavljanja trajnog zapisa, u mnogome olakšava rad sa računarom: traženje grešaka u programu je mnogo jednostavnije kada se takozvani „trejs" (trece, mod u kome računar ispisuje rezultate izvršavanja svake naredbe) mod primeni na štampaču, programi se bolje razumeju kada se pojave na papiru obzirom da se tada sagledava njihova celina, ROM se ne može razumeti ako se ne izlista na štampaču i tako dalje. Pored toga, svaki vlasnik štampača je oslobođen brige koje nastaju kada se neki danima razvijani program ne učitava sa kasete — program uvek može da se otkuca sa listinga za relativno kratko vreme.

Štampač je potreban svima onima koji žele da koriste računar za obradu teksta, pisanje pisama i slično. Za njih je posebno bitno da odaberu štampač koji daje zadovoljavajući kvalitet teksta. To se svodi na izbor tipa štampača.

Osnovni tipovi

Najjeftiniji štampači su termički. Kod njih se tekst proizvodi zagrevanjem posebnog papira na nekim mestima. Dve dobre strane ovih štampača su, pored niske cene, brz i tih rad. Loših strana je mnogo više: radi se na termičkom papiru koji docnije mora da se čuva „na suvom, hladnom i mračnom mestu" da ne bi pocrneo. Pismo pisano na takvom papiru ne možete nikome da pošaljete pre nego što ga, na primer, ne fotokopirate, što vrtoglavo povećava troškove korespondencije. Osim toga, termički štampači obično pišu na uskom papiru (najviše tridesetak slova u redu) koji je neobično skup, tako da se, nekoliko meseci posle kupovine, pokazuje da je cena papira tolika da se više isplatilo kupiti skuplji štampač. Loših osobina ima još, ali na njih ne treba trošiti prostor: kupovina termičkog štampača je, u 99% slučajeva, loša investicija.

Računari srednjih cena su tipa „dot-matrix". Kod njih se, pomoću posebne trake, proizvodi istovremeno veći broj tačaka koje docnije formiraju slova. Ovo je vrlo brz metod, radi se na običnom papiru, a proizvedeni dokument je relativno dobrog



loie ozbiljnija primena računara ne može se ni zamisliti bez štampača Jedan od najpopularnijih modela firme Seikosha

																		Interfejsi				
Naziv štampača	Cena (funti)	Vrsta	bez deformisanja slova	podvlačenja	inverzni kar.	različita vel.	blok grafika	hi-res grafika	max. slova u redu	karaktera po inču	linija po inču	logičko traženje	dvosmerno štampanje	karaktera/ sekundu	bafer	Max. širina papira (inča)	kopija	RS232C	I-EEE	20 ma	Centronics	brzina prenosa
Anadex 9500L	1070	Dot matrix	x	–	–	–	–	x	132	10	6–8	x	x	150	700	16 5	6	x	–	x	x	4800
Bytewriter	485	Daisywheel	–	x	–	–	–	–	80	10–15	4	–	–	12	12	16 5	1	–	–	–	x	????
Centronix 150	587	Dot matrix	–	–	x	–	–	–	132	5–16	6–8	x	x	150	60	10	5	x	–	x	x	9600
Centronix 353	2164	Dot matrix	x	x	x	x	x	x	132	10–16	3–12	x	x	200	4000	15	6	x	–	–	x	19000
Dolphin WM2000	454	Dot matrix	x	x	x	x	x	x	80	5–16	6–8	x	x	130	2000	10	3	x	x	x	x	9600
DMP 100	249	Dot matrix	x	x	–	–	x	x	80	5–10	6	–	–	50	80	9 5	1	–	–	–	–	4500
Epson MX80 II	506	Dot matrix	x	x	x	x	–	x	132	5–16	6–8	x	x	80	258	10	2	x	x	–	x	9600
Epson MX100	661	Dot matrix	x	x	x	–	x	x	233	razne	raz	x	x	60	258	15 5	2	x	x	–	x	9600
EGP 115	149	sa perima	x	x	–	–	–	x	80	10–20	6	–	x	12	80	4 5	1	x	–	–	x	1200
Fujitsu 830	2288	Daisywheel	x	x	x	x	–	x	158	10–12	3–8	x	x	60	256	15	6	x	–	x	x	1200
HP 62905A	699	Dot matrix	x	x	–	–	x	x	132	5–16	4	x	x	80	–	10	2	–	x	–	–	????
HP 2671A	968	Termički	x	x	–	–	–	–	132	10–16	6	x	x	120	2000	6.5	1	x	x	–	x	9600
HP2631B	3180	Dot matrix	x	x	x	–	–	–	227	4–16	1.7	x	x	180	256	15 6	5	x	x	x	–	9600
IBM 5242	2456	Dot matrix	x	x	–	–	–	–	198	10–15	1–12	x	x	160	2000	14 5	5	–	–	–	–	15000
NEC PC 8023	458	Dot matrix	x	x	–	–	x	x	136	10–12	6.6	x	x	100	256	10	1	–	–	–	x	????
OKI Microline 80	270	Dot matrix	–	–	–	–	x	–	132	5–16	6 6	–	–	80	256	9 5	4	x	–	–	x	9600
OKI Microline 82A	452	Dot matrix	x	–	–	–	x	–	132	5–16	6.6	x	x	120	132	9 5	4	x	x	x	x	1200
Olivetti 121	1200	Daisywheel	–	–	–	–	–	–	64	10–15	???	x	–	120	1000	16	2	x	x	–	x	2400
Seikosha GP100A	210	Dot matrix	x	–	x	–	–	x	80	5–16	6.6	–	–	40	80	10	4	–	–	–	x	1200
Star DP 510	300	Dot matrix	x	x	x	–	–	x	132	10–17		x	x	100	3000	10	5	x	–	–	x	9600

x=da
–=ne
?=podatak nema smisla za printer

kvaliteta. Ipak, kvalitet koji se postiže ne matričnim štampačima je ratko takav da može da se meri sa standardnim i električnim pisaćim mašinama, a u štampač ne može da se umeće već broj listova (za pravljenje karbon-kopija). Za najveći broj korisnika „dot-matrix" štampač predstavlja, ipak, najbolji mogući izbor.

Za one koji žele da im računar piše veoma važne poslovna pisma i dokumente poželjna je nabavka takozvanog „daisywheel" štampača. To je, u suštini, kvalitetna električna pisaće mašina koja je snabdevena interfejsom za povezivanje sa računarom. Tekst se proizvodi pomoću kuglice koja udara standardnu traku. Obzirom da se piše slovo po slovo, ovi štampači su vrlo spori, ali je zato kvalitet štampe neoblično dopadljiv. Većina velikih kompanija koristi „daisywheel" štampače ali su oni, za pojedinos, preskupa alternativa. Dot-matrix štampač, koji proizvodi slova ne dovoljno velikoj matrici (npr 10 17), imitira „daisywheel" štampače dovoljno verno za većinu primena.

Umanjenje slova

Sledeća karakteristika štampače na koju treba obratiti pažnju je „umanjenje slova". Jeftiniji štampači pišu sva slova tako da bivu deformisano tako da bude jednako visoko kao slovo o što unekoliko smanjuje čitljivost dobijenog dokumenta. U našoj tabeli iksovima su označeni štampači kod kojih su slova normalne veličine sa protezanjem i izvan granica reda po potrebi.

Naš je utisak, ipak, da su i štampači koji umanjuju slova dobri za većinu primene. Pismo pisano na ovakvom štampaču biće svakome razumljivo, iako na prvi pogled možda neće biti mnogo lepo. Čitanje takvog pisma kod primaoca stvara osećanje da ga je pisao računer — pa, zašto bežati od toga?

Nejednaka slova su karakteristika skupljih štampača i električnih pisaćih mašina. Kod njih je, na primer, slovo m nešto šire od slova a i slično. Ovakvi štampači daju dokumenta dopadljivijeg izgleda, ali se pomoću njih teže komponuju tabele i proračunava broj slova u redu. Čini nam se, zato, da ova osobina, koju inače hvale kao karakteristiku skupih štampača, može da se pretvori u nepogodnost za korisnika.

Podvlačenje reči

Mogućnost podvlačenja je sledeća karakteristika. Većina jeftinijih štampača ne poseduje mogućnost da podvuče neki deo teksta. Podvlačenja sa kod njih postiže štampanjem čitavog reda „–" znakova, čime redovi postaju neprirodno razmaknuti. Bolji (i skuplji) štampači mogu malo da pomere slova nagore, što omogućava da se neki deo teksta podvuče i tako istakne. Ova karakteristika je značajna kod obrade teksta.

Kao alternativa podvlačenju, neki štampači mogu da deo teksta štampaju „inverzno": slova su bela na tamnoj pozadini. Ova karakteristika nije mnogo značajna za štampače koji omogućavaju definisanje sopstvenih karaktera, a ni za ostala — dobijeni tekst izgleda pomalo „neprirodno", odnosno ne odgovara standardima na koja su nas navikla pisaće mašine i knjige.

Redefinisanje karaktera

Grafički karakteri služe za crtanje slika. Štampači su, u principu, namenjeni za pisanja teksta, ali je itekako korisno ako se pomoću njih mogu nacrtati manje ili više složene slike. Zato neki štampači omogućavaju redefinisanje karaktera. Posle redefinisanja svako slovo dobija neku drugu funkciju i postaje specijalni grafički karakter. Kod štampanja grafičkih karaktera ali minišan je prostor između linija i između slova u okviru iste linije. Ima nekih štampača postoji i grafika visoke rezolucije, koja omogućava direktno prenošenje slika sa ekrana na štampač.

Broj karaktera

Sledeći parametar pomenut u tabeli je broj karaktera u redu. Većine štampača ima 80 karaktera u redu, što je dovoljno za pisanje teksta u formatu koji imitire pisaću mašinu. Neki štampači imaju nekoliko veličina slova — u jedan red, na primer, može da stane 40 većih slova, 80 normalnih ili, po želji korisnika, 120 umanjenih, 120 slova u redu je lepa pogodnost, pogotovu za one koji nameravaju da umnožavaju (foto-kopiraju) ono što kompjuter proizveda, pa nastoje da na istoj strani stane što više teksta. Uvećana slova su neophodna za naslova, podnaslova ili poruka koje treba posebno naglasiti. Većina štampača ne omogućava kombinovanja slova raznih veličina u istom redu.

Veličina slova

Kao merilo veličine slova, u tabeli je dat broj karaktera koji se semštaju na prostor od 2,5 cm (ovo je internacionalna jedinica pošto 1 inč ima 2,5 cm) i broj linija koja staju ne 2,5 cm. Ako uzmete standardni list papira i pomoću lenjira ga podelite na kvadratiće odgovarajuće veličine, steći ćete utisak o veličini slova koje neki štampač proizvodi i videti koliko ona odgovaraju vašim potrebama.

Brzina štampanja

Sledeći parametri se odnose na brzinu štampanja. Ukoliko nam materijalne mogućnosti dopuštaju, izabraćemo printer koji štampa što veći broj karaktera u sekundi obzirom da na taj način štedimo vreme rada računara. Ipak, brzina nije previše bitna osobina: najčešće nam je bitno da računar obavi neki posao i ne gledamo da li će se on obaviti za 30 sekundi ili jedan minut. Brži štampači su skuplji i trebe dobro da razmislimo pre nego što se odlučimo za kupovinu nekog „vrlo brzog". Posebno treba da imamo u vidu pitanje da li ćemo štampač toliko koristiti da njegova brzina dođe do izražaja?

Neki štampači imaju određen stepen „inteligencije" koji se označava kao „logičko traženje" (logic seeking). Štampači opremljeni ovom mogućnošću neće, na primer, vraćati glavu za štampanje na početak sledećeg reda ako je od njih zahtevano da nešto štampaju na njegovoj sredini. Neki štampači su opremljeni mogućnošću dvosmernog štampanja: printer ispisuje red u toku povratka glava na njegov početak pišući unazad. Ovim je brzina štampanja udvostručena.

Brzina štampanja se izražava preko broja karaktera u sekundi koje štampač ispisuje. Treba imati u vidu da se glava štampače koji ne omogućava dvosmerno štampanje posle ispisivanja svakog reda mora vraćati na početak, što unekoliko umanjuje realnu brzinu rada.

Kapacitet bafera

Dok štampač ispisuje tekst, računar bi mogao da nastavi sa radom. Svaki štampač je zato opremljen određenim tzv baferom (buffer) u koji se smeštaju slova koja treba otkucati. Veći bafer omogućava računaru da radi i dok printer štampa, dok se manji bafer brzo napuni i računar mora da čeka njegovo „pražnjenja" da bi poslao nova karaktere. Veličina bafera je u tabeli data u bajtovima.

Već smo rekli da neki štampači omogućavaju pisanje nekoliko kopija. U sledeće dve rubrike tabele naveden je njihov broj uz maksimalnu širinu papira koji može da se koristi.

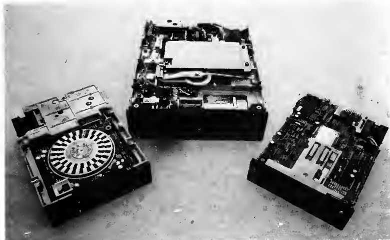
Tip interfejsa

Poslednja grupa rubrika izlaže interfejse koji su ugrađeni u pojedine štampače. RS232 je prisutan kod većine a češti su i neki drugi interfejsi. Pri kupovini štampače obratite pažnju na interfejse koja poseduje vaš računar i, ako nisu odgovarajući, izaberite neki drugi štampač ili dokupite potreban interfejs. Kod standardnih interfejsa bitno je da li računar može da šalje informacija brzinom koja je štampaču potrebna. Ova brzina se izražava u baudima (bodima) i data je u poslednjoj rubrici tabele. Ovaj podatak nije nešto što će početniku biti razumljivo. No, pogled ne tehničke specifikacije vašeg računara će omogućiti da proverite da li je printer odgovarajući. Ukoliko vaš računar ne šalje podatke traženom brzinom, savatovali bismo vam da razmislite o nekom drugom štampaču.

U tabeli nije obuhvaćen printer za ZX81 i ZX Spectrum obzirom da je to jedan od veoma retkih štampača koji je isključivo namenjeni jednom određenom računaru. Ovaj štampač piše ne specijalnoj hartiji čudnog sjaja koja, na svu sreću, ne tamni ako je izložimo svetlosti. Štampač omogućava ispisivanja čitavog sadržaja ekrana u svakom trenutku što, kada uzmemo u obzir visoku rezoluciju kojuj Spectrum poseduje, predstavlja lepu karakteristiku. Na žalost, kvalitet prenošenja slike je prilično slab, neka slova bivaju iskrivljena dok se delovi drugih praktično i ne štampaju. Obzirom da je papir uzak (semo 32 slova u redu), neprimenljiv je za „ozbiljne poslove" — teško da će se ikada odlučiti da nekome pošaljete pismo pisano na njemu. Najbolji rezultati se postižu kod teksta u kome ima „invarznih karaktera" (beli karakteri na tamnoj pozadini) a najbolji kod običnih slova. Ovaj štampač je, dakle, korisan isključivo za listinja programe koji sa nalaze u memoriji računara i slične „sitne poslove" iako mu je cena od pedesetak funti veoma primamljiva, ne smatramo da kupovina ZX printera predstavlja dobru investiciju. Spectrum je dobar računar koga, ukoliko planiramo ozbiljne primene, možemo da opremimo RS232 interfejsom i bilo kojim štampačem iz srednje klase.

disk jedinice

Pored štampača, u „jaku" opremu računara spadaju i disk-jedinice, kojima se memorija računara može proširiti do impozantnih razmera, ali za njihovu nabavku prosečan ljubitelj računara ipak ima znatno manje opravdanja. Autor ovog teksta poznaje nekoliko vlasnika računara koji su nabavili disk-jedinice „da bi sistem bio kompletan", a zatim, pritešnjeni drugim poslovima, upoznali samo neke osnovne naredbe i nastavili da koriste kasetofon za sve praktične potrebe. Disk jedinice su neobično korisne ali ako imamo dovoljno para i, razume se, ako znamo šta da radimo sa njima.



Na disketu se može smastiti ogromna količina podataka — da li ih imate dovoljno za tako enormni apetit. Disk — jedinice firme Toshiba

Disk ledinice

naziv	(funte) cena	drajvova	prečnik	Strana	Gustina	Traka	Podela	Kapacitet	Interf
Commodore 8250	1489	dva	5 25	jedne	dvostruka	154	meka	2 MB	1EEE488
Commodore 8050	1029	dva	5 25	jedna	dvostruka	77	meka	1 Mb	1EEE488
Commodore 2031	454	jeden	5 25	jedne	dvostruka	35	meka	171 Kb	1EEE488
Control data ZL141B	155	jedan	5.25	jedna	dvostruka	40	svaka	250 K	BBC
Control deta ZL142	339	dva	5.25	jedna	dvostruka	40	svaka	500 K	BBC
Control deta ZL242	454	dve	5 25	dve	dvostruka	40	svaka	1 Mb	BBC
Cumene DA8035	856	dva	5 25	jedna	dvostruka	80	meka	655 K	Apple II
Eicon FD1DD	1426	jedna	8	jedna	jednostr.	77	meka	500 K	Apple II
Eicon FD82DD	2242	dva	8	jedna	dvostruka	77	meka	2 Mb	Apple II
Megastar	1888	dva	8	dve	jednostr.	—	meka	1 Mb	Apple II
Penny Giles 6000	1483	dva	5.25	jedna	jednostr.	40	maka	262 K	RS232
Penny Giles 600B	1089	jedan	5 25	jedna	jednostr.	40	meka	81 K	RS232
Tandy Model I	349	jedan	5.25	jedna	jednostr	40	meka	175 K	TRS 80 I
Tande Model II	999	jedan	8	jedna	dvostruka	80	meka	486 K	TRS 80 II

Disk-jedinica služi za smeštanje programe i podataka — ona, dakle, ne omogućave ništa što ne bi bilo moguće i pomoću kasetofone. Međutim, disk-jedinica svoje zadatke obavlja daleko brže od svakog kasetofone, što je neobično važno za one koji koriste računar u poslovne svrhe. Da bi računer bio pravilno upotrebljen, neophodno je da obavlje poslove znatno brža nego što bi to činili ljudi, da je relativno samostalan (na zahteva, na primer, da ga stalno opslužuje operater koji bi pramotavao ka-



Jeftina varijanta disk-jedinice koja nije ispunila očekivanja: Sinklerov mikrodrajv

sete) i da što manje greši Obrada veće količine podataka pomoću kasetofona nije ekonomične — neka od razloga za ovu tvrdnju izložili smo kada smo govorili o povezivanju računara sa kasetofonom Specijalizovani kasetofoni predstavljaju neko polu-rešenje, ali postaju nedovoljni kada količina informacija koju obrađujemo postane značajna Za ozbiljne poslovne primene, dakle, disk-jedinica postaje nezamenljiva.

Disketa

Kao medij za smeštanje programa i podataka korista se takozvane diskete To su pločica pravučene posebno osetljivim magnetnim materijalom koje su podeljene na „sektore" i „staze" Na njih računar veoma brzo upisuje i sa njih čita podatka, omogućavajući praktično trenutan pristup informacijama Prenošenja programa od dvadesetak kilobajta sa kasete u memoriju računara može da zahteva desetak minuta, dok sa toliki program sa diskete učita u dalu sekunde!

Dvostruka ili jednostruka?

Najvažniji parametar dat u tabeli, pored naziva i cene, je „jednostrukost", odnosno „dvostrukost" disk-jedinica Neke disk-jedinice se sastoje samo od uređaja u koji se smešta jedna disketa, dok drugi omogućavaju istovremani rad sa dve diskete Drugi su bolji i, naravno, skuplji Za rad sa disk-jedinicom, naime, računar mora da buda opremljen posebnim programom koji se zove DOS (disc operating system). Ovaj program se sastoji iz više daiova koji se automatski unose u memoriju računara kada se za tim ukaže potreba Zato se kod „dvostrukih" disk-jedinica u jedan drajv stavlja takozvana „sistemska disketa" (disketa na kojoj je snimljen DOS), a u drugi disketa sa kojom se u tom tranutku radi Ukoliko posedujemo jednostruku disk-jedinicu, na svakoj disketi koju koristimo moraće da bude snimljen čitav DOS, koji može da odnese i 80% prostora Za naše programe i podatke ostaće, dakle, malo mesta

Prečnik je sledeći paramatar u tabeli Obično se izražava u inčima (1 inč = 2,5 cm) Većina računara koristi diskete od 5,25 inča, a u poslednja vrema se javljaju i takozvane mini-diskete prečnika od svega 3,5 inča

Neke disk-jedinice snimaju programe i podatke sa jedne, a druge sa obe strane diskete Što se ekonomičnosti tiče, disk-jedinica koja snimaju sa obe strana su bolje jer na disketu tako staja dvostruko više podataka, ali je sa druge strane, vreme traženja podataka nešto duže

Gustina pakovanja

Dalja povećanje količine informacija koje staju da disketu postiže se boljim pakovanjem podataka Jeftinija disk-jedinice pakuju podatke „normalno gusto", nešto bolje „dvostruko gušće" a najskuplje — „četvorostruko gušće" Jasno je da se na disketu koja je umeće u neki uređaj koji smešta podatke na obe strane u četvorostrukoj gustini smešta otprilike 8 puta više podataka nego kod običnog Cena ovakvog uređaja je obično tri puta veća, a vrame pristupa podacima nije mnogo duže Sva u svemu, povećanja gustine pakovanja je investicija isplativa samo onima koji planiraju izuzetno čestu upotrebu disk-jedinica

Gruba i meka podela

Već smo rakli da se disketa deli na trake i sektora Ako je uporedimo sa gramofonskom pločom, trake su spiralne formacija po kojima ide igla (u ovom slučaju specijalna glava za čitanja) Broj trake na disketi je različit i neposredno uslovljava količinu informacija koja može da se smesti na nju Što se sektora tiče, neke diskete se prava tako da se glava za čitanje pozicionira pronalaženjem nekog od većeg broja otvora koji su na njoj izbušeni (ovo se naziva „gruba podala" ili „hard sectoring") dok se kod drugih na disketi nalazi samo jedan otvor („meka podela" ili „soft sectoring") Sa stanovišta korisnika sasvim je nevažno kog je tipa disk-jedinica — dovoljno je da pazi kakve diskete kupuje Bolje disk jedinica omogućavaju ravnopravno korišćenja disketa sa „grubom" i „makom" podelom

Svi prethodni parametri su sumirani u kapacitat diskete To je broj kilobajta informacija koji može da se smesti na jednu disketu pri optimalnoj upotrebi Za većinu disk jedinica ovaj broj se mari kilobajtima (jedan kilobajt odgovara hiljadi slova ili polovini kucane strane teksta) ali se neke mogu podičiti kapacitetom od nakoliko megabajta

Sinklerov mikrodrajv

U okviru poglavlje „disk-jedinica" trebe svakako pomenuti i jedan uređaj koji, u suštini, tu ne spada Sinklerov mikrodrajv Kada se ZX Spectrum, sredinom 1982, pojavio na tržištu, Sinklar je najavio mikrodrajv za „ovu (1982) godinu" iz razloga koja možemo samo da nagađamo, mikrodrajv, je „zakasnio" skoro godinu dana — na tržište je došao tek avgusta 1983 Kao i svaka disk-jedinica, mikrodrajv služi za smeštanja veće količine informacija To, međutim, nije disk-jedinica u smislu u kom smo je do seda posmatrali — u kaseti se nalazi obična magnetna traka Zahvaljujući ugrađenom mehanizmu, traka se okreće po principu „beskonačne petlje" i programi i podaci se razmarno brzo unose u mamoriju Spectruma

Na jednu traku staja oko 80 kilobajta informacija. Prosečno vrame traženja podataka (to je vrama koja je računaru potrebno da pronađa svaki podatak) je 3,6 sekundi, a maksimalno — 7 sekundi. Između računara i mikrodrajva podaci se razmenjuju brzinom od oko 13 000 bauda, što znači da se čitava memorija Spectruma (48 kilobajta) čita za svega 3—4 sekunde

U celini gledano, mikrodrajv je skup Cena od 50 funti je varijava — da bi se priljučio na računar, treba kupiti i RS 232 interfejs koji košta 35 funti i kasete sa trakom koja koštaju 5—6 funti To znači ukupna osna ovog uređaja iznosi 100 funti — koliko i osnovne varzija Spectruma Magnetna traka koja sa brzo okreće izložena je valikoj opasnoti — pucanju U nekim angleskim časopisima prijavljene su prve nevolja sa pucanjem trake — ovo nije samo trošak već i nepnjatnost zbog gubitka 100 K programa i podataka Još je, naravno, rano da se daja kompletan sud o mikrodrajvu — u tranutku kada ovo pišemo ne amatra se da je njegova kupovina previše dobra investicija

ploteri

Ploter (crtač) je manje potrebna periferijska jedinica. Za razliku od štampača koji omogućava ispisivanje teksta, ploter je prvenstveno namenjen crtanju složenih slika. Princip na kome se zasniva ovaj uređaj je sasvim jednostavan: specijalni flomaster se kreće po papiru ostavljajući trag. Neki ploteri mogu da menjaju flomastere proizvodeći tako višebojne crteže koji su najčešće vrlo efektni

njima. Za prosečnog (naročito jugoslovanskog) vlasnika računara ploter je pravi luksuz.

Što je najgore, ploter ne može da zameni štampač. Svaki ploter, naime, raspolaže mogućnošću pisanja slova (i to raznih veličina), ali su ta slova namenjena legendama na crtežima, a ne standardnom pisanju. Obzirom da svako slovo računar crta, brzina pisanja je toliko mala da se upotreba plotera za pisanja dužeg teksta ne isplati.

U našoj tabeli prvo mesto zauzima vrata plotera. „Ravni" ploteri pišu pokretnim perom po papiru koji miruje, dok „doboš-ploteri" zadržavaju pero u istoj ravni, dok se papir okreće na specijalnom valjku-dobošu.

Sledeća rubrika prikazuje broj pera (specijalnih flomastera) sa kojima ploter može da operiše. Računar, pod programskom kontrolom, može da naredi izmenu pera u svakom trenutku i tako radi sa nekoliko boja ili nekoliko para različitih debljina. Kod plotera sa manjim brojem pera, korisnik u toku rada može da zaustavlja ploter i manuelno menja pero, ali to oduzima vreme i smanjuje preciznost crteža.

Vrlo značajan parametar pri izboru plotera je format papira na kome crta. Bezmalo svi ploteri koji su prikazani u našoj tabeli rade sa papirom formata A3, ali ima i jeftinijih modela koji rade sa A4 formatom, kao i onih koji crtaju na znatno većim listovima. Čini nam se da format A3 predstavlja dobar kompromis između potreba i cena.

Merenje brzine rada štampača je vrlo jednostavno, dok se kod plotera nije lako odlučiti za parametar kojim se može predočiti njihova brzina. Odlučili smo da navedemo koliko dugačku pravu liniju (u milimetrima) svaki ploter može da nacrta za jednu sekundu. Čim linija počnu da se „krive" brzina značajno opada, ali podaci koje smo dali mogu da posluže za rangiranje plotera. Sve u svemu, brzina ovde nije mnogo bitna karakteristika: crtanja slika obično traje dovoljno dugo da ušteda nekoliko minuta ništa ne znači. Ploter nije uređaj koji bi mogao da liferuje veliki broj kopija — najčešće se slika jednom nacrta, a docnije se umnožava foto-kopiranjem.



Idealni tehnički crtač: Ploter firme „Hewlett Packard"

Ploter je vrlo skupa periferijska jedinica. Visoka cena je izazvana činjenicom da se u njemu nalazi mnogo preciznih mehaničkih elemenata i relativno malo elektronike. Ploter nabavljaju samo oni korisnici računara koji se bave arhitekturom, građevinom i dizajnom i koji nemaju mogućnost da konste neki veći kompjuterski sistem opremljen

Ploteri

NAZIV	cena	Tip	Max. pera	Papir	Brzina (mm/sec)	Interfejsi
Waternabe WX4633	2520	ravni	10	A3	250	1EEE13749, RS232
Waternabe WX4634	2287	ravni	2	A3	250	RS232
Waternabe WX4635	2092	ravni	1	A3	250	1 EEE
Waternabe WX4671	1026	ravni	1	A3	50	RS232, 1EEE258
Watenabe 4675	1489	ravni	6	A3	50	RS232, 1EEE258
Watenabe 4731	1600	doboš	4	A3	200	RS232—105, 1—EEE347
Elcton P11	4799	ravni	31	A2	250	RS232
Scan SC14	4000	doboš	9	A3	200	1—EEE347, RS232C, RS423

palice za igru

Većina ljubitelja TV igara ima dosta iskustva sa palicama. Ona modu da se pokreću u dve dimenzije i tako omoguće igraču potpunu i jednostavnu kontrolu objekta na ekranu (teniskog reketa, na primer). Kod računara se palice obično zamenjuju nekom grupom tastera (na primer, 8 može da označava pokretanje na gore, 4 na desno, 2 na dole, a 1 dole-levo) koji, naravno, predstavljaju samo siromašnu alternativu.

Palice za igru nisu skup dodatak (najčešće koštaju desetak funti, a neki proizvođači ih čak i poklanjaju onima koji kupe štampač, disk-jedinicu i slično), a veoma su isplative, pogotovu za one koji vole video-igre. Palice ne moraju da se upotrebljavaju samo u igrama: korisnik koji treba dosta da crta na ekranu može da nabavi palice i programira računar da crta liniju koja je oblikovana pokretanjem palice. Slika sa ekrana se, docnije, može preneti na papir posredstvom štampača ili plotera.

Jedini problem sa palicama za igru je njihovo često lomljenje. Obzirom da su ovo uređaji koji treba da budu što jeftiniji,

Daleko od naših uslova. Modemi, između ostalog, mogu da obavljaju i dužnost elektronske sekretarice



Za zagrižene ljubitelje video-igara Jedan od modela palice za igru

izrada nije naročito solidna, pa palica strada u rukama temperamentnog igrača. Neki proizvođači ne zamenjuju polomljene palice, smatrajući da je do lomljenja došlo krivicom korisnika (što je, na kraju krajeva tačno), dok druge daju potpunu garanciju. Izgleda da se više isplati kupovati palice ovih drugih, pošto je veliki problem slati bilo koji uređaj u inostranstvo na opravku.

Modem

Modem je uređaj pomoću koga se više računara povezuje u telefonsku mrežu. Priključivanjem modema računar može da opšti sa drugim računarima i prenosi im programe i podatke. Modemi se najčešće koriste za povezivanje malih računara u velike sisteme: nameštenici neke veće kompanije mogu kod kuće da imaju stone računare koji će, telefonskim linijama, biti povezani sa firminim računskim centrom. Ovi računari će obavljati obradu podataka koje im programer daje i, po potrebi, opštiti sa centralnim računarom koji će im dostavljati tražene podatke i ne čije će se diskove smeštati rezultati rada.

Što se amaterske primene tiče, modem može da bude koristan ako u nekom gradu postoji veći broj ozbiljnih korisnika istog računara. Jedan telefonski poziv je mnogo jeftiniji od poštarine za slanje kasete, a posao se obavlja znatno brže. Ipak, u Jugoslaviji je broj korisnika računara koji poseduju modem minimalan, kod nas ima toliko različitih računara da prenošenje podataka između njih nije moguće ili se ne isplati.

Strategija izbora

Rekli smo da je za naše vlasnike računara greška pri izboru relativno bazbolna: pogrešno nabavljeni računar sa proda otprilike po trostrukoj ceni. Sa periferijskom opramom nije takav slučaj: ukoliko opremite računar velikim brojem skupih periferijskih uređaja, teško ćete naći kupca koji će biti spreman da odjednom plati sumu koja bi pokrila njihovu cenu. Zato vam savetujemo da najpre kupite računar i da ga upoznate. Ukoliko se uverite da sistem zadovoljava vaše potrebe, možete da počnete sa postupnom nabavkom periferijskih uređaja. Kada nabavite štampač i disk-jedinice, moraćete da se pomirite sa činjenicom da nećete biti u mogućnosti da zamenite sistem za neki moderniji u doglednoj budućnosti. Ipak, i sadašnji računari su dovoljno moćni; nema sumnje da će se u budućnosti koristiti daleko bolji sistemi, ali su i mogućnosti sadašnjih sasvim dovoljne za sve primene.





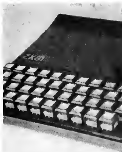
Hardverski dodaci

sitnice koje život znače

Za periferijsku opremu smo rekli da predstavlje složenu i skupu dopunu računara. Na tržištu su, međutim, i mnogobrojni jeftini moduli, proširenja, tastature, generetori tona, sintetizatori glasa i, uopšte, dodaci koji uvećavaju mogućnosti računara ili još češće, ispravljaju njegove slabosti. ZX81 (ili, kako ge neki nazivaju, „šampion u slabostima", ovo je više izraz simpatije nego kritika), na primer, ima lošu tastaturu — zašto da ne damo neku funtu i opremimo ga boljom? Ili, TRS80 nema generator tonova — zar ne bi bilo dobro da kupimo jeden dobar i tako poboljšamo računar? Ukoliko se povedete za ovakvim mislima, pali ste u bunar bez dna

Ako dodatak na koji ste „bacili oko" ima prvu funkciju, razmislite o tome da li vam je on potreban, pa ako zaključite da jeste, kupite ga — nećete pogrešiti No, ukoliko hardverski dodatak ispravija neku manu vašeg računera, moraćete ozbiljno da se zamislite Kada ste kupovali računar bili ste (ili je bar trabao da budete) upoznati sa njegovim dobrim i lošim stranama. Čin kupovina je značio da ste zaključili da su nedostaci podnošljivi A onda ste pročitali neki oglas vidali nakoliko efektnih fotografija i zaključili da vam naka osobina vašeg računara smeta Pošto je računar koji ima mana koja traba korigovati dodatnim hardverom verovatno bio prilično jeftin, dokup nove opreme će povećati njegovu cenu i to toliko da se za ukupnu sumu mogao kupiti bolji računar! Ako, na primer, zamenite tastaturu vašeg ZX81 i povećate njegovu rezoluciju, utrošili ste upravo onoliko novca koliko košta ZX Spectrum! Zar nije bilo bolje da ste prodali ZX81 i kupili Spectruma koji, uz bolju tastaturu i generator tonova, ima i mnoge druge prednosti? Dakle, ukoliko zaključite da vaš računar ima karakteristike koja vam više ne odgovaraju, bolje je da razmislite o novom računaru nego da ga kitite hardverskim dodacima!

Ova strana popunjena „prijateljskim savetima" verovatno nije dovoljna da „hakera" koji su zavoleli svoj računar odgovori od kupovine dodataka za svog ljubimca Zato smo pregledali priličan broj časopisa i popisali hardverska dodatke koji izgledaju interesantni Pomenućemo neke od njih počevši od onih koji su najinteresantniji — EPROM programatora.



Ispomoć sumnjive vrednosti Mehanička tastatura za računar ZX81

Programatori EPROM-a

EPROM je neobično popularan termin među vlasnicima računera, ali je sigurno da među našim čitaocima ima dosta onih koji za njega nisu čuli, pa moramo ukratko da objasnimo o čemu se radi Svaki računar je opremljen RAM-om, tj memorijom u koju korisnik upisuje svoja programa, i ROM-om tj memorijom u koju su trajno upisani programi koji omogućavaju normalno funkcionisanja računara RAM se briše kada isključimo računar, a ROM nikako ne možemo da menjamo Pa, postoji li mamorija u koju možemo da upišemo svoja programe tako da se oni ne izgube kada isključimo računar? Odgovor je potvrdan a takva memorija se naziva EPROM (Eresable Programmable Read Only Memory) Radi se o čipu u koji se upisuje 2, 4, 8 ili 16 kilobajta mašinskog programa koji se ne briše sve dok EPROM ne buda izložen dugotrajnom delovanju ultraljubičastog zračenja ili, da ne budemo





Razmislite dobro pre nego što podlegnete iskušenju Mehanička tastatura i optička olovka za Spectrum



Korisno pomagalo za ljubitelje crtanja Optička olovka

previše stručni, dok ne preleži nekoliko sati na suncu. Programi ne mogu da se upisuju u EPROM bez specijalnog uređaja koji se zove EPROM programator.

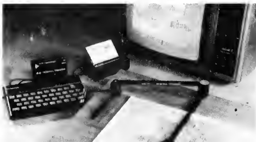
EPROM programator, u principu, može da bude nezavisan od računara. Ipak, mnogo je pogodnije da se on poveže sa računarom pošto je lakše kucati mašinske programe preko mnemoničkih skraćenica nego pomoću heksadekadnih (ili, da stvar bude teže, binarnih) brojeva. Za rad sa EPROM programator vrlo je poželjno da posedujemo računar opremljen statičkim (CMOS) memorijama. Jedan od poznatih uređaja ovog tipe, namenjen računarima PET i BBC, proizvodi firma **Computer Interface Design, 4 Albert Road, Margate, Kent CT9 5AM, England.** Cena uređaja je umerena — 46 funti.

Šta da radimo sa EPROM-om kada smo ga već isprogramirali? EPROM-i su sastavni deo mnogih uređaja u kojima se nalaze procesori — to mogu da budu alarmni sistemi, mikrotalasne rerne ili „inteligentni" roboti. Ipak, malo nas ima prilike da konstruiše ovakve uređaje pa nas najčešće zanima primena EPROM-a kod računara. U EPROM možemo da smestimo neke često korišćene programe koji će na taj način postati sastavni deo našeg kompjutera. Ukoliko, na primer, posedujemo TRS 80 i radimo sa mašinskim jezikom, dosadiće nam da neprekidno učitevamo program koji se zove editor/asembler. Ako posedujemo EPROM programator, smestićemo ovaj program u EPROM i on će nam stalno biti na raspolaganju.

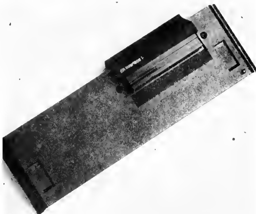
EPROM-i su vrlo pogodni za računare koji poseduju statičke memorije, ali ne mogu direktno da se uključe u kompjutere sa dinamičkim (dinamičke memorije imaju jeftiniji modeli kao što su ZX81 i Spectrum). Za ove računare potrebno je dodati nešto hardvera da bi EPROM mogao da dođe do izražaja. Kada je reč o Sinclairovim računarima, ovaj hardver proizvodi firma **Solidisk Technology Ltd, 87 Southamouth Park Road, Sothend-on-Sea, Essex, England.** Uređaj se prodaje za 16 funti i pod imenom XROM System i ugrađuje se u svaki ZX Spectrum. Posle ugrađivanja, u njega mogu da se priključe EPROM-i i ROM-ovi sa programima korisnika ili komercijalno nabavljeni moduli, jedan takav je uračunat u cenu uređaja.

Biperi i ostale zujalice

Firma Solidisk Technology Ltd proizvodi i druge interesantne dodatke. Ukoliko vam je dosadilo to što nemate osećaj da li ste pritisnuli neki taster na vašem ZX81 ili niste, možete da kupite takozvani BEEP unit (samo dve funte) koji se oglašava tonom svaki put kada pritisnete neki taster. Ako vam se ne dopada izgled vašeg ZX81 ili Spectruma, možete da nabavite novu kutiju u koju ćete postaviti računar. Kutija je opremljena profesionalnom tastaturom i cenom od 50 funti. Ako vam smeta niska rezolucija ZX81, možete da kupite tzv. „80 Column Card" koja će omogućiti pisanje 80 slova u redu i neprekidni rad u FAST modu (cena je „samo" 50 funti). Ukoliko vam je, najzad, dosadio vaš ZX81, firma Solidisk Technology Ltd. će biti srećna da vam proda neki drugi računar.



Prenošenja crteža na ekran. Uređaj za digitalizaciju slike nazvan digitalni traser za računar Spectrum



Veza sa spoljašnjim svetom. Intefejs 1 (RS 232) omogućuje povezivanje Spectruma sa mikrodrajvom kao i sa nekim profesionalnim štampačima

Sintetizator glasa

Jeste li nekada poželeli da komunicirate sa Spectrumom rečima? To će vam omogućiti firma **DK Tronica, Unit 2, Shire Hill Ind. Est, Saffron Waldam, Essex CB1 3AQ** sintetizatorom glasa za ZX Spectrum koji košta samo 40 funti. Kupovinom ovog uređaja vaš Spectrum će progovoriti — ne samo da će moći da izgovori određeni broj standardnih engleskih reči već ćete moći da ga programirate tako da razgovara sa vama! Iste firma prodaje i analizator zvuka (pomoću ovog uređaja Spectrum bi trebalo da bude u stanju da razume neke reči koje izgovorite), ali u ovaj uređaj ne treba imati previše poverenja. Optička olovka je još jedan od proizvoda firme DK Tronica. Pomoću nje na ekranu možete da crtate slike koje će biti registrovane u memoriji računara, nećete, dakle, morati ručno da unosite koordinate tačake koje treba osvetliti. Optička olovka se priključuje u port na poleđini svakog Spectruma (poželjno je, ipak, da posedujete model sa 48 K RAM-a) i košta 20 funti.



Poslednja novost: Interfejs 2 omogućuje proširenje ROM-a Spectruma igrama, kao i uslužnim, obrazovnim i sistematskim programima; za sada se mogu kupiti samo ROM-ovi sa igrama, među kojima je većina dobro poznata i našim ljubiteljima računara, ali se uskoro očekuju i ozbiljni programi

Mehanička tastatura

Najveću zaradu firma DK Tronics ostvaruje na prodaji kutija za Spectruma. Jedna ovakva kutija je dovoljna da se u nju smesti čitava štampana ploča Spectruma se nakim hardverskim dodacima. Zašto da zatvaramo Spectruma u kutiju? Zato što se na njoj nalazi profesionalna tastatura sa odvojenom grupom tastera za brzo kucanje brojeva i, po želji, RS232 interfejs. Cena osnovne verzije ove kutije je oko 50 funti, ali treba upozoriti potencijalne kupce da je za instaliranje Spectruma u nju potrebno potpuno raskiopiti ovaj računar i zalemiti neke kontaktke. Za one koji se ne usuduju da ovo urade, ista firma prodaja posebnu tastaturu koja se povezuje se portom na zadnjoj strani Spectruma bez ikakvih drugih intervencija. Cena je, jasno, nešto veća — oko 70 funti

Rezervna centrala

Stoni računari, u načelu, rade samo dok su povezani sa mrežom. No, firma **Adaptors & Eliminators Ltd, 14 Thamas St. Louth, Lincs, England** je verovatno bila obaveštena o predstojećim restrikcijama struje u Jugoslaviji pa je proizvela BATTPACK. Ovaj uređaj se povezuje sa ZX81 (sada postoji i verzija za Spectruma) i omogućava nekih 30 minute rada koji ne zavisi od gradske mreže. Ako, dakle, dok ste pripremali neki složen program nestane struje, ostaje vam dovoljno vremna da snimite program na kasetu i tako ga „spasete". Iskreno se nadamo da kod nas neće biti restrikcija i da firma Adaptors & Eliminators Ltd neće zaraditi mnogo novca od Jugoslovena!

Ubrzano učitavanje

Mnogo je kritika upućeno načinu na koji ZX81 snima programe na kasetu: upis je užasno spor i nepouzdan, ne postoji mogućnost verifikacije programa, česte su greške... Samo onaj ko poseduje računar zna koliko nerviranje izaziva situacija u kojoj tek napisani program neće da se učita sa kasete. Firma **PSS, 112 Oliver Street, Coventry CV6 5FE, England** za 18 funti prodaje hardversko/softerski dodatak nazvan QSAVE. Q dolazi od quick pošto, uz pomoć ovog dodatka, ZX81 može da snimi čitav sadržaj memorije za samo tridesetak sekundi. Uređaj pruža mogućnost verificiranja snimljenih programa i, ako je verovati proizvođaču, eliminiše sve probleme pri učitavanju. Na „Galaksijinom" konkursu za najbolji domaći program nagrađen je jedan izuzetan program tipa QSAVE koji bez ikakvog dodatnog hardvera povećava brzinu učitavanja osam puta i, uz to, obezbeđuje Funkciju VERIFY.

4 džungla u računaru

Komercijalni programi

Sistemski programi

maternji i ostali kompjuterski jezici

Svaki računar je opremljen jednim ogromnim sistemskim programom na koji vlasnik obično ne obraća pažnju bejzik interpretetorom. Ovaj program je smašten u ROM računara i omogućava njegovo normalno funkcionisanja Kada, na primer, uključimo računar, on izvršava nekoliko desetina naredbi koje omogućavaju da se na ekranu pojavi dobro poznati kurzor. Bejzik interpratator se provlači kroz sve što radimo

Računar, ipek, može da funkcioniše i bez bejzik interpretatora On se „isključuje" unošenjem programa koji omogućave upotrebu nekog drugog programskog jezika, rad sa tekstom, mašinskim jezikom ili upotrebu računara u nake druge svrhe Programi koji daju računaru neka potpuno nove mogućnosti spadaju u kategoriju sistemskog softvera

Sistemski programi su u najvećem broju slučajava izuzetno složani Pisani au na mašinskom jaziku I prostiru se na desetak kilobajta memorije Kada govonmo o sistemskim programima u širam smislu, u njih ubrajamo I neke mala programe koji dodaju računaru naku novu mogućnost ili naredbu, kao i programa koji kontrolišu periferijske uređaja Ipak, prvi deo ovog poglavlja je posvećen novim programskim jezicima

Bejzik je, u najvećem broju slučajeva, sasvim dovoljan za svakog korisnika Njegova mogućnosti, pogotovu ako se radi o boljem bejziku, dovoljna su za sva ono što korisniku može da „padna na pamet" Ipek, bejzik je programski jezik koji se interpratira, što znači da se programi pisani na njamu izvršavaju razmerno sporo. Neki ovakve probleme rešavaju pišući programa na mašinskom jaziku, ali ja takvo rešenja previše komplikovano za naiskusna vlasnika Za njih je izlaz nabavka sistemskog programa koji računaru omogućava rad sa nekim bržim programskim jezikom kao što su paskal ili fort.

Jezik za matematičare

Paskal ja viši programski jezik namenjen uglavnom matematičarima Nastao ja kao sintaza nekih svojstava algola, fortrana i bejzika (prva dva programska jezika su dalako starija od bejzika i koriste se mahom na većim računarima) Svaki programar koji zna bejzik lako će da piše programa na paskalu paskal poseduje praktično sva bejzik naredbe, iako su naka od njih praimenovane (CALL umesto GOSUB i slično) No, paskal ima i mnoštvo naredbi koja su bejziku nepoznate daklarisanja promenljivih I nizova, pozivanje potprograma uz prenošenje argumanata, rekurzivno pozivanja potprograma i slično Za neiskusnog vlasnika poslednja rečenice nije previše jasna Za njega ja jedino bitno da je paskal programski jazik koji se pravodi To znači da korisnik najpra kuca čitav program, ispravlja greška pri kucanju i smešta program napisen na paskalu na kasetu Zatim koristi posebnu naredbu koja nareduja računaru da ovaj program praveda na mašinski jazik Posla nakoliko trenutaka od korisnika se zahteva da uključi kasetofon na koji se snime mašinski program Taj program može docnija da se koristi kao svaki drugi mašinski program, što znači da se izvršava 10—100 puta brže od odgovarajućeg bejzik programa Paskal, dakla, predstavlja izvanrednu alternativu bejziku za ona koji žela da sastavljaju programe koji će se brzo izvršavati Iako ja namanjan matematičarima, ovaj jazik pruža valika mogućnosti za sastavljanja igara Ipak, ukoliko sa odlučite na nabavku paskal kompajlera, traba da pripazita na nakoliko sitnice

Paskal kompajlar je valiki program koji zauzima dober deo mamorije računara Ukoliko vaš računar ima mamoriju od 16 kilobajta, vrlo je varovatno da čitav program na može da stana u nju Zato su se autori programa pobnnuli da ga podale u nekoliko dalova Da biste uneli program, na primer, morate da učitate takozvani „editor" Po unošenju programa traba učitati „kompajlar" kako bi program bio prevedan na mašinski jezik Sledi testiranje programa i, naravno, pronalaženje grešaka Kada pronađate grešku, ponovo učitavate editor da biste je ispravili I kompajlar da prevedete ispravljanu varziju Ukoliko je program složen, neprakidno unošenja različitih programa odnosi toliko vremana da se rad se paskal kompajlerom ne isplati Pre nago što, dakla, kupite ovakav program, raspitajte sa da li ja memorija računara dovoljna da ga pnmi

Osim toga, naki programi na prevoda paskal na mašinski jazik, nago na takozvani „p kod" Program preveden na „p kod" se izvršava gotovo jednako brzo kao i mašinski, ali ja pra njegovog startovanja sa kasete neophodno unati poseban program koji kontroliše njegov rad To praktično znači da se napisani program nikada ne moža izvršavati samostalno, što ga često čini neprimenjivim

„Pascal kompajlar" za ZX Spectrum ja isplativa investicija Programi prevedani pomoću njega izvršavaju se, zavisno od operacija koja sa koriste, četrdesetak puta brže od odgovarajućih bejzik programa Primere radi, program koji generiše hiljadu slučajnih brojeva a zatim ih reda u rastući niz koristeći Šelovu sortirku na bejziku radi dvadeset, a u paskalu jedva pola minuta Cena ovog programa u originalnoj varziji ja čitavih 25 funti, dok se od domaćih praprodavaca može nabaviti za 800 dinara

Fort, fortran i kobol

Fort je ralativno novi programski jezik Znatno je nižeg nivoa od bejzika (što ja jezik „primitivniji", razumljiviji ja kompjuteru i manja pristupačan korisniku) I po mnogo čemu je sličan RPN notaciji koju koriste Hewlett-Packardovi džepni računari Ovaj jazik ja znatno brži od bejzika i poseduje neke nove mogućnosti, kao što ja semostalno definisanja naredbi (korisnik moža da oformi specijalne naredbe koje rešavaju neke njegove probleme) Ovaj jazik ja namenjen mahom računarima manjih mogućnosti kao što je ZX81 I pored činjanice da se fort dalako lakše pravodi od paskala, u 16 kilobajta mamorija ZX81 na moža da stane čitav program za korišćenja ovog jazika. Zato se u toku rada neprakidno unosi editor, kompajlar i drugi uslužni programi što, kada se uzma u obzir mala brzina kojom ZX81 učitava programe sa kaseta, čini čitav jazik teško upotrebljivim I pored toga, fort interpratator sa prodaje za čitava 32 funte, varovatno zato što je uz njega priključena i knjiga uputstava od sedamdesetak strana Kod domaćih praprodavaca fort moža da se nabavi za 300 dinara — dovoljno malo da se odlučita na kupovinu

Za računare veće memorije I boljih karakteristika teško možemo da preporučimo fort njagovo razumevanje zahteva dosta napora, a primana teško pruža nešto više od paskala Fort ja, ipak, preporučljiv za one koji još nemaju smelosti da se upuste u programiranje na mašinskom jaziku, ali bi želali da upoznaju jazike nižeg nivoa, kao i za one koji su nakada posedovali Hewlatt-Packardove džapne računara i žela da se podsete „starih vramena".

Čak i najskupiji računar, sem po sebi, ne ume da radi ništa korisno. Da bi se upotrebio za rešavanje problema od značaja za njegovog vlasnika, računar mora da bude opremljen odgovarajućim programima. Ako je vlasnik računara dobar programer, on će te programe moći da sastavi samostalno, čime će na najbolji mogući način da zadovolji svoje potrebe. Dosta je, međutim, onih koji nemaju znanja ili vremena da pišu programe koji su im potrebni. Za njih rade grupe inženjera organizovanih u kompanije za proizvodnju programa (softvera). Ovakve kompanije procenjuju potrebe korisnika i sastavljaju programe koje mogu da ih zadovolje. Komercijalni programi se uvek odlikuju velikim mogućnostima i ogromnim brojem opcija koje treba da ih učine prilagodljivim potrebama najšireg kruga ljubitelja računara. Kupujući ovakav program, plaćamo i za ono što nam nije potrebno, ali i obezbeđujeme neke buduće potrebe koje još ne možemo da predvidimo.



Ništa bez dobrih programa. Programska podrška predstavlja jedan od najvažnijih činilaca u uspehu jednog računara

Za računare opremljene jakom periferijskom opremom (najvažnije su disk-jedinice) pišu se fortram i kobol kompajleri. Fortram je programski jezik velikih kompjuterskih sistema. Kao što mu ime govori (**FOR**mula Translation ili prevođenje formula), namenjen je matematičarima i poseduje moćne naredbe za rad sa kompleksnim brojevima, povećanu tačnost, obilje matematičkih funkcija i druge pogodnosti. Za igre je sasvim nepogodan jer ne omogućava praktično nikakvu kontrolu ekrana. Dobri fortran kompajleri koštaju više stotine funti i, za sada, nisu prilagođeni Spectrumu.

Kobol je programski jezik namenjen poslovnoj obradi podataka. Programi pisani na njemu su vrlo slični eseju pisanom na engleskom — umesto standardnih znakova matematičkih operacija koriste se reči i slično. Ovaj jezik je izuzetno dobro opremljen naredbama za rad sa periferijskim uređajima i formiranje datoteka. Nema svrhe nabaviti kobol kompajler ako vaš računar nema bar stotinak kilobajta memorije i nije opremljen dobrim štampačem i disk-jedinicama. Cena kobol kompajlera prevazilazi sve pomenute i može da dostigne 1000 funti. Ovakav program nije napisan za Spectruma, a vrlo je verovatno da nikada neće ni biti!

Šta radi asembler . . .

Sa posebnih programskih jezika preći ćemo na korišćenje onoga što je skriveno u samom računaru — mašinski jezik. Unošenje mašinskih programa na način koji je pomenut u delu posvećenom iskusnim programerima je teško prihvatljivo za ozbiljan red. Pri pisanju programa i ispravljanju neizbežnih grešaka mnogo je lakše operisati sa mnemoničkim skraćenicama nego sa heksadekadnim (ili, još gore, binarnim) brojevima. Probleme ovoga tipa rešavaju takozvani asembler programi.

Dobar asembler omogućava korisniku da, najpre, unese mašinski program pomoću mnemoničkih skraćenica i snimi ga u izvornom obliku na kasetu. Zatim korisnik kuca specijalnu naredbu kojom otpočinje prevođenje programa na mašinski jezik i unošenje prevoda u memoriju računara počevši od proizvoljno odabrane lokacije. Ukoliko u toku prevođenja računar naiđe na neku skraćenicu koja ne postoji ili čiji je adresni deo pogrešan, korisnik može jednostavno da je ispravi. Kada je čitav program preveden na mašinski jezik, primenom nove naredbe prevod se smešta i na kasetu za docniju upotrebu. Mašinske programe je teško pisati i greške su česte. Dobar asembler program treba da omogući korisniku da sa trake ponovo učita izvorni program, unese ispravke i ponovo ga prevede na mašinski jezik. Asembler program se, dakle, nalazi u memoriji računara dok mašinski program ne bude potpuno testiran i ispravljen, posle čega se rezultat snima na traku, a asembler-program briše.

Neki asembler-programi pružaju i dodatnu mogućnost: izvršavanje mašinskog programa naredbu po naredbu. Na ovaj način korisnik je u mogućnosti da locira i ispravi greške u mašinskom programu na razmer-

no bezbolan način: izvršavanjem programa naredbu po naredbu pronalazi se mesto na kom registri procesora imaju neočekivani sadržaj, proučava se taj sadržaj, pogleda njegov izvor i — greška je nađena.

. . a šta disasembler

Disasembler je program koji obavlja inverznu funkciju. Često, naime, kupujete neke druge komercijalne programe pisane na mašinskom jeziku. Ako želite da ih analizirate i prilagodite nekim svojim posebnim potrebama, neophodno je da program koji se nalazi u memoriji prevedete na jezik mnemoničkih skraćenica i nađete mesta u programu gde dolazi do grananja kako biste uveli nove opcije. Ovakve probleme rešavaju disasembleri. Najpre se u memoriju unese disasembler, a zatim mu se naloži da se kasete prihvati mašinski program koji treba analizirati. Računar najpre utvrđuje na koje se memorijske adrese unosi ovaj program, unese ga i počinje njegovu analizu. Na ekranu (i štampaču, ako je potrebno) se pojavljuje program prikazan preko mnemoničkih skraćenica, a zatim i tabela sa naredbama uslovnog i bezuslovnog grananja. Disasembler obično prevodi program na takav način da se prevod može snimiti na kasetu i docnije prepravljati pomoću asembler programa. Zato je važno kupiti asembler i disasembler od iste firme — na taj način se ostvaruje kompatibilnost.

Zašto je DEVPAK najbolji?

Kod ZX 81 i ZX Spectruma asembleri i disasembleri se unekoliko razlikuju od uobičajenih. ZX 81 na raspolaže mogućnošću smeštanja mašinskih programa na kasetu, pa je neophodno da se mašinski program „maskira" u bejzik. To se najčešće postiže pomoću REM naredbi koje su za druge vlasnike računara sasvim nepotrebne — u okviru njih se, normalno, smeštaju komentari kao što je naziv programa, ime njegovog autora i slično. Uz malo veštine, međutim, u okviru REM naredbi može da se smesti čitav mašinski program, koji bejzik docnije jednostavno poziva. Ovo smeštanje na sebe preuzima asembler, dok disasembler obavlja inverznu funkciju — „čita" REM naredbe i prevodi ih u oblik prihvatljiv za korisnika. Disasembler, osim toga, omogućava i traženje određenog niza znakova kroz čitavu memoriju, postavljanje takozvanih „prekidnih tačaka" u programu (kada računar, u toku izvršavanja mašinskog programa, naiđe na „prekidnu tačku" on umesto da nastavi sa izvršavanjem programa prestaje sa radom i na ekranu prikazuje sadržaje procesorskih registara), kao i disasembliranja ROM-a računara. Asembler i disasembler za ZX81 koštaju dvadesetak funti, a u Jugoslaviji mogu da se nabave za oko 1000 dinara. Postoji više verzija asemblera i disasemblera za Spectrume. Preporučili bismo vam, ipak, DEVPACK. Ovaj paket obuhvata dva programa i mnogobrojne opcije koje uključuju sve ono što je spomenuto u tekstu (izvršavanje programa naredbu po naredbu, mešanje programa u memoriju sa programom na kaseti i slično). DEVPACK je nešto što svaki ozbiljniji vlasnik Spectruma treba da nabavi, cena od 14 funti je umerena, a program, zajedno sa



Namenski programski jezici: Obilje specijalnih programskih jezika nema nikakve veze sa ratovima proizvođača — oni postoje samo zato da vam pomognu *da se u svakoj prilici na najbolji način sporazumete sa svojim računarom*

uputstvima, može da se nabavi i od jugoslovenskih preprodavaca za svega 500 dinara.

Mi smo vam, dakle, preporučili DEVPACK, ali koji DEVPACK? Da bi stvar bila složena, isti autori su pripremili tri verzije koje su numerisali brojevima I do III. Ne, razume se, iz inata, nego da poboljšaju svoj program. Sva tri programa slede istu koncepciju i sva tri su dobra, ali su novije verzije opremljene kompletnijim spiskom komandi i, što je najvažnije, imune su na „greščice" kojih je bilo u programu DEVPACK I. Rekło bi se da je najbolje kupiti DEVPACK III? Ne možemo da budemo sasvim sigurni u to jer program nismo videli, ali je DEVPACK II imao jednu nepogodnost za početnika: uputstvo se sastojalo od skripti za upotrebu programa DEVPACK I i dodatka namenjenog programu DEVPACK II. Početnik, kome je dovoljan problem da razume termine kao što su „source file", „object file", „origin" i slični mora najpre da čita jedno uputstvo, a onda da zaboravlje nešto od onoga što je pročitao ili, da stvar bude gora, da menja upravo stečene navike. Ukoliko sada nabavljate program, kupite DEVPACK III, ali ako ste se već navikli na DEVPACK I ili DEVPACK II — ne trošite novac. Program koji imate je dovoljno moćan za vaše potrebe.

Jednostavne rutine

Ima i mnogo manjih sistemskih programa. Zadržaćemo se na dve pisana za Spectrum. Jedna od mana ovog računara je što ne omogućava da se prenumeriše program. Brojevi naredbi, naime, najčešće nemaju linijska brojeva koji su pravilno raspoređeni i koji omogućavaju jednostavno kucanje programa. Zato su korisnici sastavili programe poznate pod imenom „renumber" (prenumerator) koji ređaju linije tako da njihovi brojevi budu 10, 20, 30 i tako dalje. Neki preprodavci programa za mala sume prodaju neke ovakve programe, često ne ističući njihovu veliku manu: oni ne menjaju adrese u GOTO i GOSUB naredbama, čime prenumerisani program postaje potpuno beskoristan. Postoje, međutim, i kompletni programi za prenumerisavanje ali, na žalost, samo na stranom tržištu. Jedan takav program prodaje Malve Ltd. za 7 funti.

Za ZX81 napisano je zaista mnogo programa, od kojih su neki izuzetno zanimljivi. Bejzik programi pisani za ZX81 mogu bez problema da rade na Spectrumu, ali je za njihovo unošenje potreban listing i dosta vremena za kucanje. Kao pomoć onima koji na kasetama poseduju mnogo programa za ZX81, napisan je program „Slow loader". Pomoću njega se automatski unosi i prevodi svaki program pisan za ZX81 — dovoljno je uneti kasetu i startovati program. Nekoliko minuta docnije program će biti preveden i smešten u memoriju ZX Spectruma. Čini nam se da je 10 funti (400 din kod preprodavaca) previše novca za ovaj program. Naime, ZX81 je spor računar, pa su dobri programi pisani uglavnom na mašinskom jeziku, sa kojim „Slow-loader" ne može da izađe na kraj.

Sa bejzika na mašinac

Za kraj poglavlje posvećenog sistemskim programima ostavili smo one najpotrebnije: poznati su kao „M-Coderi", „Code Translatori", „MCC programi" ili, kratko, bejzik kompajleri. Ovi programi omogućavaju automatsko prevođenje bejzik programa na mašinski jezik. Time je razvijanje samog programa olakšano jer se bejzik lako ispravlja, a izvršavanje ubrzano prevođenjem na mašinski jezik. Bejzik kompajleri na prvi pogled ukidaju svaku potrebu za drugim programskim jezicima, učenjem mašinskog jezika i drugim razmišljanjima o ubrzavanju računara. No, to nije uvek tako. Pre svega, da bi se sastavio *savršen* kompajler (tj. kompajler koji će moći da prevede sve bejzik programe na mašinski jezik), potreban je računar solidne memorije opremljen disk jedinicama. Bejzik kompajler za TRS 80, na primer, prevodi sve programe na mašinski jezik, čime se njihovo izvršavanje ubrzava 10—40 puta, ali zahteva memoriju od 48 kilobajta, dve disk jedinice i — 500 maraka za kupovinu programa!

Većina bejzik kompajlera, međutim, ne može da prevede sve naredbe. Veliki problem nastaje kod prevođenja naredbi koje zahtevaju rad sa brojevima koji nisu celi, pa se ove naredbe često izostavljaju. FOR--NEXT petlje izazivaju nove probleme, pa se i od njih često odustaje. Naredbu po naredbu, bejzik postaje sve siromašniji, da bi se kod kompajlera prilagođenih računarima skromnijih mogućnosti (prvenstveno ZX81 i Spectrum) sve završilo na relativno malom broju naredbi. Kompajler za Spectruma, nazvan „Super C Compiler", prevodi svega 42 naredbe i omogućava rad jedino sa celim brojevima. Dakle, nema govora o tome da se ovaj program koristi za prevođenje svih bejzik programa. Međutim, može da se dopusti jedan kompromis: ukoliko nam je potreban program koji treba brzo da radi, odreći ćemo se mnogih bejzik naredbi i pisati program tako da bude pogodan za kompajliranje. Kada ga napišemo, testiramo i ispravimo sve greške, prevešćemo program i dobiti mašinsku verziju daleko jednostavnije nego da smo odmah pisali program na mašinskom jeziku.

Sa Super C Compilerom, međutim, ima problema drugog tipa. Njegovi autori su se toliko uplašili mogućnosti da neka od njihovih mušterija počne da komercijalizuje programe prevedene uz pomoć njihove rutine da su kompajler sastavili tako da dobijeni mašinski kôd smesti u memoriju i

omogući njegovo izvršavanje ali ne i snimanja na kasetu. Ograničenje je za dobrog programera besmisleno, ali autori Super C Compilera verovatno računaju sa tim da je dobar programer dobro upućen u tajne mašinskog jezika pa mu ovaj program nije ni potreban. Za početnika je bolje da kupi program M Codar (cena je otprilike ista), koji ima neke prednosti i nedostatke u odnosu na Super C Compiler ali je manje „zaštićen"

Kompjuterske poliglote

Teško je izabrati dobre sistemske programe. U svakom slučaju, treba kupovati samo ono što nam je potrebno. Ako počnemo da se opremamo svim postojećim jezicima, verovatnoo je da ćemo bacati novac, jer nećemo naći vremena da ih sve naučimo, a ako ih i budemo naučili, videćemo da ne pružaju mnogo više od bejzika. Asembler i disasembler su, kao što smo rekli, nužni za svakog boljeg programera i treba ih dopuniti nekim monitor-programom ako takav postoji. Monitor program će nam omogućiti da steknemo uvid u ono što se dešava u samom računaru, shvatimo raspored sistemskih promenljivih, čitamo zaštićene programe i slično.

Postoje mnogobrojni uslužni sistemski programi koji predstavljaju proširenja bejzika nekim novim naredbama. Ovi programi mogu da budu neobično korisni, ali najčešće ostaju sasvim neupotrebljeni: onima koji ne poseduju disk-jedinice zamorno je da svaki put kada uključe računar učitavaju proširenja bejzika, osim toga, programe pisane na „proširenom bejziku" ne mogu da daju prijateljima koji nemaju potreban sistemski program. Sve u svemu, sistemski programi su prilična investicija bez koje može da se prođe. Ipak, sigurno je da će uvek biti neobično popularni. Poznajući psihologiju vlasnika računara, dolazimo do zaključka da je najteže odoleti iskušenju pri kupovini sistemskih programa: njihove mogućnosti, bar na papiru, predstavljaju upravo ono što je našem „elektronskom ljubimcu" preko potrebno.

Uslužni programi

mogu li da vam pomognem?



Veliki kompjuterski sistemi se najčešće kupuju opremljeni sistemskim programima, a zatim se nabavljaju mnogobrojni uslužni programi bez kojih sistem ostaje potpuno neiskorišćen. Kod kućnih računara situacija je nešto drukčija: zbog obilja igara kućni računar može da bude iskorišćen čak i kada ne radi ništa „korisno". Ipak, postoji i grupa uslužnih programa koji u mnogome mogu da povećaju praktičnu vrednost vašeg računara.

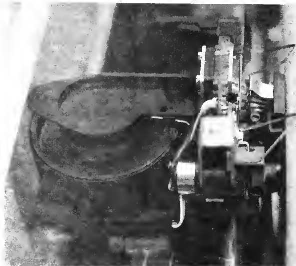
Umesto pisaće mašine

Tekst-procesor je neophodan svakome ko ima potrebu za pisanjem nekih tekstova. Ti tekstovi mogu da budu knjige, stručni članci, ali i najobičnija pisma i trebovanja. Za korišćenje tekst-procesora treba imati računar sa najmanje 32 kilobajta memorije (poželjno je da on ima profesionalnu tastaturu) i štampač, dok je disk jedinice korisna, ali ne i neophodna. Tekst-procesor eliminiše potrebu za pisanjem koncepta, višestrukim prekucavanjem teksta da bi se greške ispravile i docnijim „ručnim" dopisivanjem i prepravkama. Primenom tekst-procesora najpre unosimo prvu verziju teksta koja se smešta u memoriju računara. Kada smo otkucali tekst, jednostavnim pritiskanjem nekoliko tastera dovodimo na ekran pojedine njegove delove da bismo ih menjali. Možemo lako da izbacujemo slova, reči, rečenice i pasuse, da umećemo nove reči ili nova slova i da više puta čitamo napisano. Čitav tekst ne moramo da štampamo odmah — ukoliko želimo da na njemu radimo docnije, jednostavno ćemo ga snimiti na kasetu (ili disketu) i isključiti računar. Sačuvevši, na primer, konačnu verziju neke knjige, nema nikakvog problema da izvršimo ispravke pri pripremanju sledećeg izdanja i ponovo odštampamo prepravljeni tekst uz minimum utrošenog vremena.

Kada je tekst otkucan, pristupa se njegovom štampanju. Dobri tekst-procesori omogućavaju korisniku da izabere broj slova u redu, širinu margine, broj redova na strani i mnoge druge parametre. Ukoliko je potrebno, računar može da izravna desnu marginu teksta, što obezbeđuje znatno dopadljiviji izgled dobijenog dokumenta. Računar, ako je potrebno, numeriše strane, izdvaja poglavija krupnijim naslovima, crta podele u tabelama i obavlja mnoge poslove koji su do skora zahtevali skupe usluge daktilografa i tehničkih crtača. Sve u svemu, tekst-procesor predstavlja izuzetno isplativu investiciju.

Bolji tekst-procesori omogućavaju i kontrolu spelovanja, koja je posebno značajna za one koji pišu tekstove na engleskom. Računar, naime, poseduje mali rečnik od nekoliko hiljada često korišćenih reči, koga korisnik dopunjava terminima koje on koristi u praksi. Kada računar naiđe na reč koja je vrlo slična nekoj iz njegovog rečnika ali ipak nije identično napisana (na primer, jedno slovo je drukčije), on na ekranu ispisuje poruku kojom korisniku daje na znanje da postoji mogućnost da je u toj reči sintaksna greška. Korisnik može da je ispravi ili, ako se radi o drugoj reči koja se piše slično nekoj koja je poznata računaru, da ignoriše upozorenja. Zahvaljujući Vukovom pravopisu, kontrola spelovanja za nas nije naročito značajna, niko uostalom, ne piše tekst-procesore za obradu teksta pisanog na našem jeziku. One koji bi želeli tekst-procesor sa mogućnošću kontrole spelovanja treba upozoriti da je za njegovu primenu najčešće potrebno posedovati disk-jedinicu: rečnik je preveliki da bi se stalno nalazio u memoriji računara.

Izbor dobrog tekst-procesora je izuzetno složen problem. Praktično jedini način da izaberete onaj koji vam odgovara je da ih probate nekoliko. Pri tome treba obratiti posebnu pažnju na to koliko se dugo izvršavaju neke komande. Ako, na primer primetite da je računaru potrebno desetak sekundi da se pripremi za naknadno ubacivanje slova u već napisanu reč, zaboravite taj tekst-procesor — on je pravljen tako da zauzima što manje memorije i, prema tome, nije dovoljno praktičan. Ne treba previše da brinete ukoliko odmah ne shvatite sve komande koje tekst-procesor razume: primena kompjutera je u mnogo čemu različita od primene pisaće mašine i potrebno je izvesno vreme da se na nju naviknete. Ukoliko tekst-procesor ima mnogo naredbi, ne treba da se zabrinjavate: primena nekoliko je sasvim dobar početak, ostale ćete upoznati spontano, radeći sa programom. Pri isprobavanju tekst-procesora neophodno je jedino da proverite sve naredbe i ispitate da li su brze i praktične. Naredba koja se upotrebljava samo jednom (na primer, naredba za porav-



Štampač vrhunskih karakteristika Ozbiljna primena računara kao što je na primer, obrada teksta ne može se ni zamisliti bez dodatne opreme

navanje desne margine i štampanja teksta) može da se izvršava i duže, ali naredbe za umetanja i brisanje teksta moraju da rade praktično trenutno!

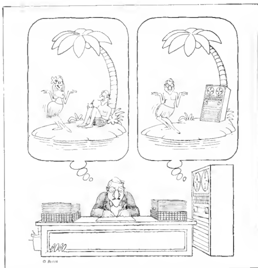
Neki tekst-procesori nisu obični programi koji se unose sa kasete ili diska već se prodaju u obliku modula Modul se direktno prikijučuje u računar (u neki spoljni port ili podnožja na samoj štampanoj ploči) I ne zahteva utrošak RAM-a — čitava memorija je slobodna za smeštanje teksta ROM za obradu teksta je vrlo dobra ideja ali ima i dve mane skuplji je od programa na traci (cena može da bude 30—70 funti pa i više) i nije moguće „udruživanje sredstava" sa nekim prijateljem koji bi dovalo do toga da do programa dodete upola cene Dobra strane su brojnije veći prostor je korisno upotrebljen, program se daleko brže izvršava od onoga koji bi se nalazio u memoriji, nema potrebe da svaki put učitavamo ogroman program sa kaseta, broj opcija je daleko veći i tako dalje

ZX81 i Spectrum nisu baš stvoreni za obradu teksta Pre svega, tastatura je daleko od prefesionalna što značajno usporava pisanje i prepravke, memorija ZX81 i neproširenog Spectruma od samo 16K može da primi premalo teksta, akran omogućava ispisivanje samo po 32 karaktera u redu, štampač piše na uskom papiru koji, da stvar bude gora, ima neprirodan srabrni izgled. To, ipak, ne znači da za ove računare, pogotovu za Spectrum, nije napisano nekoliko dobrih takst procesora Sudeči prema prikazima koji su nam bili na raspolaganju, najbolji se zove „Tasword II", a prodaje ga firma Tasman Software, 17 Hartley Crescent, Leeds LS6 2LL za skromnih 14 funti Program omogućava, pored ostalog, i kontrolu 64 slova u redu! Njegovi autori su, očigledno, shvatili da je 64 minimalan broj slova u redu koji omogućava normalan rad (standardni red otkucan na pisačoj mašini ima upravo toliko slova) i bili su spremni da plata neprijatnu cenu slova se neobično uska (sastoje se od samo tri tačka po horizontalnoj dimenziji), što znači da su previše stilizovana i da ih je, bar dok se ne naviknemo, teško prepoznavati Ovu manu donekle kompenzuju velike mogućnosti kontrola i formatiranje izlaznog dokumanta na štampaču kao i mnogobrojne opcije koje omogućavaju promene i pomeranja teksta, traženja određenih reči i njihovu eventualnu automatsku promanu i slično Tasword II predstavlja značajno unapređenja prvog tekst-procesora koji je programirala ista firma i koji se i dalje prodaje za 8 funti

Kućne banke podataka

Sledeća popularna grupa uslužnih programa omogućava rad sa grupama podataka (datotekama) Ozbiljan rad sa datotakama je teško moguč bez disk-jedinica i besmislen bez štampača Uz disk-jedinice se dobija takozvani DOS program koji zavisno od svoje složenosti, daje manji ili veći broj naredbi za rad sa datotekama Ipak, ove naredbe su više namenjene primeni u okviru složenijih programa nego manuelnom traženju i smeštanju potrebnih podataka Ovakvi programi, pre svega, omogučavaju jednostavno kreiranje datoteke i određivanja izgleda svakog njenog sloga Zatim je omogućeno jednostavno unošenje podataka, ispravka, snimanje i čitanja podataka na kasetu ili disketu i, što je najvažnije, traženje potrebnih podataka.

Pre nego što nabavimo uslužni program za rad sa datotekama, treba raalno da procenimo mogućnosti našeg računara za takvu primenu Memorija, pre svega, treba da bude dovoljno velika da ne zahteva neprekidno unošenje podataka — najmanje 32 kilobajta Štampač je praktično neophodan jer predstavlja jedini način da dobijemo neki trajni zapis o onome što smo radili Za amaterske primene (obrada manje količine informacija, na primar „pamčenja" indeksa tekstova obuhvačenih u nakom časopisu čuvanje manjeg adresara i telefonskog imenika ili praćenje brojeva koji su u poslednjih stotinak kola izvučeni na lotou) kasetofon može da posluži kao zamena za disk jedinice Ako nameravamo da vodimo računovodstvo nekog objekta male privrade ili da iznajmljujemo usluge našeg kompjutera, teško možemo da očekujemo da će obrada podataka biti ekonomična bez disk-jedinice Treba, osim toga, da reaino pro-


© Botts

cenimo našu potrebu za obradom veće količine informacija Možda želimo da se snaodemo ovakvim programom da bismo okolini pokazali da je računar na koji smo potrošili dosta novca korisno upotrebljen? Prema našim iskustvima, iako je računar i stvoran za obradu podataka i organizovanje datoteka značajan broj korisnika nema potrabe za ovakvim radom što ne znači da oni ne mogu da upotrebe računar u druge svrhe

Program za rad sa datotekama sa procenjuje prama brzini kojom razvrstava podatke i pronalazi željene informacije Organizovanje optimalnog pretraživanja i sortiranja datotake nije ni malo jednostaven programarski zadatak ovoj temi su posvećene ogromna knjige Sa stanovišta korisnika bitno je da računar može da pronade traženi podatak bar nekoliko puta brža nego što bi to uradio čovek koristeći dobru kartoteku. Ukoliko, na primer, na traci držimo telefonski imenik sa 500 brojava telafona i primatimo da je računaru potreban minut da pronade neki od njih, možemo slobodno da odustanemo od tog programa za rad sa datotekama — obična sveska će ga lepo zamaniti No, ukoliko računar obavi traženje za sekundu program je verovatno dobro napisan

Dok je ZX81 prespor i poseduje pramalu memoriju za rad sa datotekama Spectrum ovde moža da nađe veču primenu Ovaj računar, doduše na radi praviše brzo, ali se to ne primečuje ako je program pisan na mašinskom jeziku Memoroja od 48 kilobajta, pogotovu ako je dopunjana mikrodrajvom, može da primi više podataka nego što ćemo u praksi imati prilike da obradujemo Tastatura predstavlja ozbiljan problem (kucanja podataka je prilično zamorno) ali — ništa nije savršeno Za Spectruma ja napisano nekoliko programa za rad sa datotekama, od kojih ja nejpoznatiji VU FILE Kod njegove primane korisnik najpra odlučuje kako slog datoteka traba da izgleda, a zatim ga popunjava podacima Traženje podataka ja prihvatljivo brzo čak i kada je memorija praktično popunjana Ovaj program je kompatibilan sa štampačem (postoji čak i poseban mod za štednju papira), ali ne podržava mikrodrajv Cena mu je 9 funti (u Jugoslaviji može da se dobije za 100 dinara što znači da ga preprodavci mnogo ne cene) i za taj novac sa, pored programa, dobijaju i relativno kompletna uputstva Nedavno se pojavio i naslednik ovog programa koji omogućava opštenje sa mikrodrajvom i nudi nove opcije objašenjane u knjizi uputstva od preko 100 strana Cena ovog programa je znatno veća (20 funti) i, u doba kada ovo pišemo, program nije na raspolaganju u Jugoslaviji

Preko ,,Galaksijinog" kluba programera se, za svega 200 dinara, može nabaviti program Turbo 1 koji su sastavili Ivan Gerenčir i Milan Pavićevič Ovaj potpuno mašinski program omogućava relativno brz pristup podacima, ispis na ekranu ili štampaču u formatu 51 karakter po liniji (slova su nešto uža nego obično ali nisu deformisana), upotrabu domačih latiničnih slova **č, ć, ž** i **š**, sortiranja podataka po abecedi i čuvanja na traci uz razne varijante snimanja i učitavanja Program ima neka prednosti i nedostatke u odnosu na strane komercijalne programe prednosti su naša slova i veći broj karaktara u redu, a nedostaci relativno sporo sortiranje i nemogućnost istovremenog sortiranja po različitim slogovima Program ja nagrađen drugom nagradom na nedavno završenom Konkursu za najbolji domači program za kućni računar i dobio je broj **ZXSP90K5003** u našem Katalogu

Obrada taksta i rad sa datotekama su oblasti u kojima mnogobrojni korisnici mogu da primenjuju računar Postoje, međutim, i usko specijalizovani programi za obavljanje nekih specifičnih poslova Zajednička karaktaristika ovih programa je enormno visoka cena Razloge za to nija teško pronaći ovakvi programi zahtevaju rad programara koji je usko-stručan za neku oblast, a takvog nija lako naći Pored toga da bi se posao isplatio, potrebno je dobiti određeni novac, ukoliko je malo potencijalnih kupaca, svaki od njih mora da plati visoku cenu Obzirom da ovakvi programi omogučavaju značajne uštede smanjivanjem personala koji vrši obradu podataka korisnicima se isplati da u njih ulože značajne sume novca Ove „značajne sume mogu da dostignu i nekoliko desetina hiljada maraka autor ovog taksta se seca jednog programa za TRS 80 koji je omogućavao vođenje poslova jedne zubarska klinike i koštao 14000 maraka ili po trenutnom kursu preko 500 000 dinara!

Jasno je da je za kupovinu ovako skupog programa potrabno najpre dobro proceniti njagove realne mogućnosti i izračunati za koliko se vramena on moža isplatiti Za to ja neophodno videti progrem na delu što će svaki prodavac sa zadovoljstvom omogučiti Obzirom da uslužnih programa ima toliko da se teško mogu nabrojati i da je svaki od njih interesantan samo za mali broj čitalaca (verovatno je da kod nas nema nikoga ko bi bio spreman da plati ovako velike sume za jedan program), nećemo se na njima zadržavati Potencijalnim kupcima preporučujemo da prate oglasa u specijalizovanim stranim časopisima

Igre
hajde da se igramo

Igre, bez sumnje, spadaju u najpopularniju kategoriju komercijalnih programa. U početku se smatralo da one predstavljaju neophodnu odskočnu dasku u svet računara. Ubrzo je, međutim, postalo jasno da one baš nisu tako važno oruđe u popularizaciji ideje o računarima kao što se u početku činilo. Njihov izbor je toliko raznovrstan a profesionalni kvalitet toliko visok da kod većine ljubitelja računara obeshrabruju kreativni rad

Iako to može izgledati čudno, izbor dobrih programa za igre zaista predstavlja pravi problem. Igara ima mnogo, svaka ima drukčiji naslov, a mnoge su, u osnovi, vrlo slične. Iz šturih oglasa može samo da se sazna da je igra koja se reklamira „najbolja" što, priznaćete, nije dovoljan podatak. Pravila igre su objašnjena vrlo maglovito — uglavnom je dato samo nekoliko opštih napomena. Ukoliko redom počnemo da kupujemo programe za igre, deset po deset funti će predstavljati ogromnu sumu, koja je svakako mogla bolje da se utroši.

Tipologija kompjuterskih igara

Postoje, u osnovi, četiri tipa kompjuterskih igara. Prvi i najnezanimljiviji omogućavaju igru nekoliko igrača uz kontrolu računara. Ove igre su neinteresantne jer u njima računar igra veoma dosadnu ulogu — mogao bi da ga zameni i jedan od igrača koji bi bio sudija. Nema nikakvog smisla da četiri igrača koriste računar kao tablu za igru „čoveče, ne ljuti se" — originalna tabla je znatno jeftinija, a računar ne pruža ništa više od nje osim osećaja otuđenosti igrača.

Druga grupa su logičke igre na čelu sa šahom. Kod ovakvih igara igrač se bori protiv računara težeći da svojom logikom i strategijom parira računarevim mogućnostima da ispituje veliki broj poteza. Nije lako sastaviti dobar program za neku od ovih igara, što većina korisnika ne može da razume.

Treća grupa su akcione igre. U njima se na ekranu brzo dešavaju raznorazne stvari i igrač mora da ispolji dobru koordinaciju očiju, mozga i ruku. On pritiska nekoliko tastera, pokušavajući da umakne od neke opasnosti i da ostvari neki cilj. Igre ovoga tipa ćete lako prepoznati po tome što se reklamiraju kao „arcade-action games" i što su praktično uvek pisane na mašinskom jeziku.

Naposletku, postoje igre-avanture. U njima su pred igrača stavljeni neki složeni zadaci koje on na početku igre najčešće samo maglovito shvata. Od njega se očekuje da se probije kroz neki začarani zamak (ili neki deo galaksije prenatrpan meteorima što je u osnovi isto), uništi određen broj duhova i đavola (ili vasionskih monstruma), oslobodi princezu i pronađe neko blago. Ove igre su, kao što ćemo videti, najveći potrošač korisnikovog slobodnog vremena i nepresušni izvor nerviranja.

Logičke igre

Logičke igre su obično vrlo stare. One su vekovima predstavljale poligon za suočavanje igrača i merenje njihovih sposobnosti koje su često poistovećivane sa inteligencijom. Poslednjih tridesetak godina računari su se ozbiljno umešali u ovu oblast — njihove velike kombinatorne mogućnosti su ih učinile nadmoćnim protivnicima u nekim igrama, ali je formalizovano razmišljanje pokazalo i mnogobrojne slabosti u odnosu na ljudski način razonovanja koji uključuje i intuiciju. Bilo kako bilo, dosta je onih koji su neverovatno zainteresovani za veštačku inteligenciju i logičke igre, za njih su stoni računari većih mogućnosti predstavljali pravo otkrovenje.

Kako treba da izgleda dobar program za neku od logičkih igara? Njegova glavna osobina mora da bude promenljivi ishod partije. Kao što je neinteresantna igra u kojoj računar neprekidno pobeđuje čoveka (primer takve igre je „nim" odnosno „gomilice") tako je za najveći deo korisnika (izuzetak su oni koji žele da dokažu svoju vrednost pobeđujući kompjuter) neinteresantan program koji će oni neprekidno pobeđivati (na primer, program za go). Da bi, dakle, logička igra bila zanimljiva, i čovek i računar moraju da imaju jednaku priliku da partiju reše u svoju korist. Obzirom da se većina logičkih igara igra na nekakvoj tabli, računari na ekranu u toku igre moraju da prikažu sliku koja će je manje ili više uspešno, zameniti. Neki programi omogućavaju da ta slika bude vrlo efektna, da figure budu u nekoliko boja i vrlo slične klasičnima. Ovo je lepo, ali ne treba da vam odvuče previše pažnje, za svakog korisnika treba da bude daleko važnije kako će program da igra nego kako će izgledati slika na ekranu. Ukoliko, dakle, naiđete na program koji crta divnu šahovsku tablu i uz to može da izgovara poteze i komentare, dobro razmislite o njemu — vrlo je verovatno da je kvalitet igre žrtvovan spoljnim efektima.

Volite li šah?

Šah je verovatno prva logička igra koja je zaintrigirala one koji stvaraju veštačku inteligenciju. Prvi šahovski programi su napisani pedesetih godina ovog veka i mogli su da budu ravnopravni protivnici jedino drugim šahovskim programima. Povećavanjem brzine rada računara i (donekle) pronalaženjem novih algoritama računari su počeli da pobeđuju „pacere" da bi u poslednje vreme postali dostojni protivnici šahistima ranga majstorskih kandidata. Poznajući napore koji se ulažu u razvoj šahovskih programa, smatramo da u potpunosti stoji tvrdnja čuvenog američkog stručnjaka za veštačku inteligenciju Minskog da će „1990 godine računar biti svetski šahovski prvak".

Treba, ipak, reći da na vrhunskom nivou mogu da igraju samo veliki sistemi ogromljeni džinovskim programima koji, uz neophodne tabele, zapremaju nekoliko megabajta memorije. Ovi programi, najčešće, nisu nepromenljivi — u toku igre stiču određena iskustva koja automatski nalaze svoja mesta u okviru postojećih tabela, čime program, kako prolazi vreme, igra sve bolje. Publici su pristupačni takozvani „šahovski računari", odnosno mini-kompjuteri projektovani specijalno za igranje šaha. Uz procesor koji je često standardan (Z80A ili, u novije vreme, 8085), ovi računari su opremljeni velikim ROM-om i dodatnim elementima kao što su čipovi za kontrolu displeja i (kod boljih modela) sintezu glasa. Obzirom da ove naprave ne znaju ništa osim da igraju šah, daleko se više isplati ogramiti neki stoni računar opšte namene dobrim šahovskim programom. No, dobar šahovski program zaista nije lako naći.

Za gotovo svaki računar iz niže i srednje klase napisan je barem po jedan program. Ukoliko su za vaš računar napisana dva programa, obavezno nabavite onaj koji je napisan docnije, obzirom da ga niko neće napisati tako da bude lošiji od postojećeg. Izbor šahovskog programa, dakle, i nije bog zna kako težak. Od interesa su poređenja šahovskih programa pisanih za različite računare obzirom da, verovali ili ne, postoje oni koji biraju računar prema šahovskom programu! U Holandiji je oktobra 1982 održan turnir u kome su se takmičili komercijalni šahovski programi pisani za VIC 20, Sargon II (čuveni program koji je za ovu priliku prilagođen za Apple II), Commodore 8032 i TI99/4A (program Chess Master). Svaki računar je sa svakim igrao po dve partije, a konačna tabela je data redosledom kojim su računari nabrajani. VIC 20 je sakupio svih 6 poena, Sargon II 3,5, Commodore 2,5 i TI99/4A — ni jedan poen. Velika mana ovog turnira je što u njemu nije učestvovao ZX Spectrum za koji je napisano čak šest interesantnih šahovskih programa. Na turniru održanom krajem prošle godine u Londonu učestvovalo je 12 šahovskih računara, među kojima se našao i ZX Spectrum. Konkurencija na ovom turniru je bila vrlo jaka i Spectrum je zauzeo pretposlednje mesto sa svega 1,5 poenom od 11 mogućih. Na turniru je nadmoćno pobedio specijalizovani šahovski kompjuter La Regance (košta 400 funti) a drugo mesto je zauzeo program Advanoe 2 4 nešto manje cene. U ovoj konkurenciji Spectrumu nije bilo mesta, što ne znači da šah-programi za njega nisu interesantni. Naprotiv!

Za Spectrum je napisano toliko progrma za šah da smo u velikoj opasnosti da propustimo neki od njih. Prvi je jednostavno prilagođenje odgovarajućeg programa za ZX81 novom računaru. Ovaj program su verovatno nabavili svi ZX-ovci u Jugoslaviji obzirom da, kod naših preprodavaca, već duže vreme košta samo 250 dinara (originalna cena 12 funti). Program na ekranu prikazuje šahovsku tablu na kojoj su figure obeležene slovima. Poseduje sedam nivoa igre: dok na nultom nivou računar odgovara praktično trenutno, četvrti nivo odnosi 5—15 minuta po potezu, a peti i šesti su namenjeni dopisnom šahu. Mnogobrojne opcije omogućavaju rešavanje dvopoteza, traženje „saveta" od računara, promenu pozicije, vraćanje poteza i drugo. Program igra solidno poštujući sva pravila igre, uključujući i „anpasant". Mana programa je što ne poznaje remije — igra do „golih kraljeva", pa i duže, čak i kada između dve



Pomoć ili stranputica: Programi za video igre mogu biti beskrajno zabavni, ali u suštini, odvlače ljubitelje računara od stvaralačkog rada i ozbiljnih primena

izmene protekne više od 50 poteza. Kao protivnik ne može da pruži ozbiljniji otpor solidnom šahisti koji nema nikakvu kategoriju, „paceri" često gube dok, svesno ili nesvesno, ne pronađu način kako se kompjuterski programi pobeđuju — posle toga program služi samo kao demonstracija mogućnosti računara.

Chess i je prevaziđen program Chess II koji, začudo, ima velika delova pisane u bejziku. Uprkos tome, računar brzo igra koristeći ovaj program tako da vreme razmišljanja na prvih pet nivoa retko prelazi minut po potezu. Od noviteta, interesantno je istaći mogućnost da korisnik traži savet računara na višim nivou od onoga na kome računar igra, tako će kompjuter lako pobediti samog sebe!

Sledeći šah je nazvan Voicechess obzirom da računar izgovara svaki potez (na engleskom, naravno) i ponekad daje neke komentare. Pažljivi čitalac će se setiti da nismo preporučivali programe koji troše prostor na sporedne fenomene kao što je sinteza glasa. Naše iskustvo govori da je Chess II bolji „igrač" od Voicechessa, iako poslednji može da posluži kao bolje demonstracija mogućnosti računara. U inostranstvu oba programa koštaju po 10 funti, dok je u Jugoslaviji Voicechess više cenjen: 150 dinara prema 100 za Chess II.

U poslednje vreme pojavila su se dva nova programa „Turk" i „Cyrus Chess". Prvi treba da dočara pravu partiju prikazujući na ekranu igračev i računarev časovnik, tako da ste stalno u strahu da ne izgubite partiju zbog „pada zastavice". Na prvim nivoima, kada se igra brzopotezno, retko ko može da se suprostavi računaru koji razmišlja stotinak milisekundi po potezu, dok na višim nivoima igra (koji dopuštaju više vremena za razmišljanja) počinju da se pokazuju mana programa Turk.

„Cyrus Chess" je, po našoj i opštoj oceni, najbolji šahovski program do koga se može doći. U početku program deluje frustrirajuće jer umesto da kucamo poteze u obliku E2E4 moramo da pomerimo kurzor na početno polje, a zatim na polje na koje figura treba da dođe. Posle izvesnog vremena, međutim, uviđamo da ovakav način kucanja poteza u mnogome smanjuje mogućnost „nesporazuma". Program omogućava brzo i jednostavno vraćanje poteza i mnoge druge opcije, ali mu je najbolje osobina što igra daleko solidnije od svih pomenutih. Nije ni čudo: „Cyrus Chess" je dvostruki prvak Evrope u konkurenciji stonih računara. Čini nam se, ipak, da ni jedan od ovih programa ne može da pruži dovoljan otpor prosečnom šahisti koji poseduje minimum iskustva u sličnim mečevima. dovoljno je igrati tiho dok se ne dođe u završnicu, a onda računar praktično stoprocentno gubi.

Otelo i bekgemon

Pored šaha, vrlo su popularni „othello-reversi" i „bekgemon". O obe ove igre već je pisano u „Galaksiji" (o bekgemonu u „Galaksiji" 102, a o „otelo/reversi" u „Galaksiji" 129) pa nećemo trošiti prostor na ponavljanje pravila. Radi se o stravnim igrama za dva igrača u kojima računar predstavlja dostojnog protivnika čoveku. Program za Spectrum igra „otelo/reversi" izuzetno dobro, pobeđujući pri tom najveći broj „protivnika od krvi i mesa". Na ekranu biva nacrtana čitava tabla i na njoj se pojavljuju potezi. Nivoi igre su ovde postavljeni više da se igrač ne bi kompleksirao: na prvih nekoliko nivoa korisnik praktično i ne primećuje da računar „razmišlja" — odgovori dolaze trenutno, ali se snaga protivnika razlikuje: na nultom nivou se još nekako i može pobediti, ali na četvrtom — nikako.

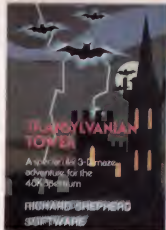
Bekgemon je igra u kojoj se bacaju kockice, što znači da hazard igra određenu ulogu. Računar igra praktično optimalno, ne rizikujući ništa. Čovek može da pobedi čak i ako nema dovoljno sreće tako što će rizikovati da bi ostvario neki cilj. Obzirom da računar „ne rizikuje", iako je moguće da čovek vrlo ubedljivo dobije partiju iako sam nema utisak da je igrao posebno dobro igran protiv računara, bekgemon nije naročito zanimljiva igra, verovatno zato ni jedan od preprodavaca programa nije smatrao za uputno da na njega potroši 12 funti (verzija za Spectrume).

Iako logičkih igara ima još dosta, ni jedne od njih nije dostigla popularnost šaha. Nabavka većeg broja programa ovog tipa teško može da bude dobra investicija — kupite program za igru koju najviše volite. Ovaj program će biti sasvim dovoljna demonstracija mogućnosti vašeg računara pred prijateljima!

Na braniku čovečanstva

Akcione igre su najinteresantnije za mlađe korisnike računara i verovatno najbolji način da nekoga navedete na kupovinu računara. Akcione igre se, naravno, ne vezuju samo za komercijalne mikroračunare — u salama mnogih bioskopa može da se zabavlja gomilom igara ovoga tipa uz obavezno ubacivanja nekoliko novčića pre početka svake partije. Mogu da se nabave i specijalizovani mini-računari koji igraju neku od ovih igara ili set sa sličnim mogućnostima. Ukoliko naiđete na nekoga ko troši značajna sredstva na flipere i kupovinu specijalizovanih kompjutera-igračaka, znajte da ste naišli na budućeg vlasnika računara i, zašto to ne reći, budući izvor besplatnih programa za akcione igre. Zato je uz računar neophodno, u demonstracione svrhe, posedovati nekoliko efektnih akcionih igara čak i ako vas one previše ne interesuju.

U akcionim igrama najčešće ste u poziciji komandanta baze koja brani Zemlju od napadača iz svemira. Napadača obično ima mnogo, ali se zato vaš brod bolje kreće i poseduje jače oružje. Akciona igra traje praktično neograničeno — na mesto napadača koje ste uništili dolaze neprekidno novi i to sve opasniji dok konačno ne budete uništeni. Cilj je igrati što duže i postići što veći skor. Za to je potrebno imati vanredne reflekse, savršenu koordinaciju nekoliko čula i pomalo sreće — prosečnom igraču je potrebno par dana da se privikne na novu igru i počne da postiže vrhunske





Dve među stotinama: Ekran u video igrama ne izgleda ni iz daleka tako raskošno kao omotnice na kasetama

rezultate. Takvog igrača nije teško prepoznati na sastanku kompjuterskog kluba: zakrvavljene beonjače i grozničavo dobovanje po stolu su izvanredan putokaz da je među vama vlasnik nove akcione igre!

Često smo govorili da računar koji radi sa televizorom u boji za većinu primena nije mnogo bitan. Za akcione igre, međutim, kolor-računar je vrlo poželjan obzirom da crno-bela treperava slika izuzetno nepovoljno utiče na oči i izaziva nesnosne glavobolje. Računar sa generatorom tonova će dati igri novu dimenziju, ali i, vrlo verovatno, izluditi igračeve ukućane.

„Space invaders" (svemirski osvajači) je verovatno najstarija akciona igra po kojoj su nastale mnoge druge. Igrač upravlja zemaljskom stanicom za odbranu od uljeza iz vasione. Specijalni laserski top može da se kreće samo po liniji koja se nalazi u dnu ekrana i pokriven je štitnicima. Uljeza ima mnogo i napadaju u nekoliko redova, neprekidno trepećreći. Igrač pomera svoj top levo-desno, puosjući između štitnika i težeći da uništi neprijatelje. Oni, za uzvrat, bacaju bombe na njega. Ove bombe postepeno uništavaju štitnike i igrač dolazi u sve veću opasnost. Igra se završava kada se prvi red uljeza spusti na Zemlju (dno ekrana) ili igračev top bude pogođen određeni broj puta.

Ako je program dobar, povremeno se dešavaju neke nepredviđene stvari: pojavljuje se, na primer, komandni neprijateljski brod koji poseduje ogromnu razornu moć, nailaze specijalno pokretljivi neprijateljski lovci, dolazi do diverzije u igračevoj bazi . . . „Space invaders" je umereno zarazna akciona igra koja predstavlja dobar test: ako vam se ona ne svidi, verovatno će i druge akcione igre za vas biti neinteresantne. Ukoliko je zavolite, niste imali sreće: potrošićete dosta novca i užasno mnogo vremena na akcione igre! Vlasnici Spectruma će doći do verzije igre „Space Invaders" ako nabave programe „Orbitar" i „Ground Attack" (Napad na zemlju).

Za Spectrum je napisana jedna posebna igra ovoga tipa koja se naziva „Time Gate" (Vremenska vrata). U njoj nije dovoljno da se krećete u dvodimenzionalnom i trodimenzionalnom prostoru, već ste prinuđeni da odete u daleku prošlost, na sam početak vremena, i da uništite prvog neprijatelja koji se uopšte pojavio! Obzirom da ova igra zahteva ne samo dobre reflekse nego i određenu strategiju, toplo bismo je preporučili. Cena od 7 funti je umerena, naročito ako se zna da program može da se nabavi i za 200—300 dinara.

Trka na život i smrt

Sledeća velika grupa akcionih igara obuhvata tzv. potere. U klasičnoj verziji igrač se nalazi u lavirintu u kome ga progone dva neprijatelja. Cilj je kretati se što duže i izbeći neprijatne susrete sa progoniteljima. Postoje i neke druge verzije ove igre: lavirint može da bude zamenjen asteroidnim pojasom, a gonioci mogu da budu neprijateljski nuklearni projektili. Možete da se nađete i u lavirintu vremena (u različitim trenucima na istim lokacijama u prostoru se nalaze ili ne nalaze neki objekti), a progonitelji mogu da budu neki istorijski događaji koje treba predupredití. U nekim verzijama svi ovi problemi mogu da se pojave zajedno (primer je igra SACTENS pisana za BBC mikroračunare), a u drugima sve ovo može da bude uprošćeno na jednostavno prelaženje prometnog autoputa.

Kada izabirate igru-poteru, treba da pazite da ne uzmete neku previše tešku (kao što je SACTENS) ni neku previše laku (kao što je Crossing za Spectrum). Vlasnicima ZX81 i Spectruma možemo da preporučimo igru „3D Monter Maze" (čudovište u trodimenzionalnom lavirintu). U njoj je igračev cilj da izađe iz kuće-lavirinta koja se sastoji od priličnog broja soba i hodnika sa raskrsnicama. Kada bi lavirint bio nenastanjen, do izlaza bi se još nekako i dolazilo, ali u njemu živi strašni monstrum. Ova kreatura je, na sreću, prilično pospana, ali ima ružnu osobinu da se probudi baš kada mu se mala i nemoćna figura koja predstavlja igrača približi. Tada nastaje trka na život i smrt — čak i kada zavarate monstruma (koji se tada umori pa opet zaspi) potpuno izgubite orijentaciju i morate da počnete potragu iz početka. Čak i ZX81 koji ima vrlo skromne grafičke mogućnosti vrlo živopisno crta monstruma (posebno njegove čeljusti) i daje izvanrednu sliku lavirinta i zidova koje vidite. „3D Monster Maze" košta oko 6 funti (verzije za ZX81 i Spectruma koštaju praktično isto) a od naših preprodavaca može da se dobije za samo 200 din.

Vazdušni letači u fotelji

Dolazimo do sledeće velike grupe akcionih igara: simulatora letenja. Većina vlasnika računara nikada nije imala priliku da sedne za komande nekog aviona, a posebno ne da u ruku uzme komandnu palicu Boinga 747! Uz investiciju od desetak funti možete da nabavite program koji će da omogući veoma vernu simulaciju leta. Na ekranu će se pojaviti prizor koji vidite kroz prednji i bočne prozore, dok ćete na njegovoj donjoj polovini videti mnogobrojne instrumente: merač brzine, visinomer, indikator potrošnje goriva, žiroskop . . . Vaš zadatak je da uzletite sa aerodroma i sletite na neki drugi. Zvuči jednostavno? Autor ovog teksta je utrošio priličan broj časova pokušavajući da uzleti sa zamišljenog aero-



Više zabavno nego korisno. Slika ekrane iz nekoliko igara na računaru TRS 80 color

droma uz pomoć stonog računara TRS 80 i jedino što je uspeo je da se razbija posle odlepljenja od piste! Leteti nije teško (program obično nudi i opciju· let bez poletanja), a sletanje je prava propast. Program za TRS 80 ima i opciju namenjenu „boljim pilotima" — pritisak na jedan taster objavljuje rat. Igrač treba da dođe do nekog protivničkog aerodroma i da ga bombarduje, izbegavajući pri tom neprijateljske lovce. Kako li je tek to?

Simulator letenja za TRS 80 je pisan relativno davno, u poslednje vreme su napisani mnogo bolji programi. Jedan od njih je prilagođen i popularnom Spectrumu. Zahvaljujući grafičkim mogućnostima, ekranu u boji i generatoru tonova, simulator je vrlo efektan. Jednostavnom komandom igrač na ekranu može da vidi mapu čitave oblasti iznad koje leti zajedno sa svim repernim tačkama. Reperne tačke se „vide" i kroz prozor (ekran) kada im se dovoljno približimo, a računar simulira čak i sam čin približavanja. I ovaj program omogućava igraču da odvojeno vežba poletanja, let i sletanja. Njegovi autori su bili miroljubivo raspoloženi obzirom da nije moguće izvoditi ratne dejstva, ali to ne znači da je program lak za upotrebu — razbijanje je rezultat svakog pokušaja da se uzleti ili sleti ukoliko za komanda ne sedne profesionalac.

Previše složeni simulatori letenja nisu naročito zanimljivi programi obzirom da je zaista potrebno učiti da bi se problemi koje oni postavljaju pred igrača rešili, ali su na raspolaganju i mnogobrojne igre koje predstavljaju pojednostavljene simulatore leta kombinovane sa „akcionim igrama". Prema tvrđenjima engleskih časopisa, najpopularnija igra za ZX81 je „3D Defender" (Branilac). U ovoj igri treba da poletite sa površine Zemlje i preseretnete flotu napadača. Vaš brod je u izvesnoj meri sličan lovcima koje koriste ratnici sa „krstarice Galaktika". Obavljanje ovoga zadatka je neuporedivo lakše od poletanja aviona, pa čak i početnik može da uništi nekoliko neprijateljskih brodova pre nego što nastrada. U ovoj igri je najvažnije upoznati pokazivanja radara i navići se na brzo reagovanje kada se protivnik nađe „na nišanu". Treba, osim toga, voditi računa o trenutnom položaju broda obzirom da može lako da se razbija o Zemlju. Moramo da priznamo da nas igra „3D Defender" nije previše impresionirala i da nismo sigurni da je vredno za nju platiti 10 funti.

Još jedna izuzetno efektna igra ovoga tipa je „Penetrator". Program je pisan za ZX Spectrum koji postavlja pred igrača nekoliko ciljeva: treba, u osnovi, prevesti avion sa jednog na drugi kraj nekog planinskog masiva. Kada startujete program, radićete sa prvim nivoom igre u koma je pejsaž koji treba da preletite razmerno jednostavan. Da bi stvar bila teška, sa zemlje na vas polEću rakete, a na vama je da pazite da se ne sudarite sa njima. Rakete koje su ispod vas možete da bombardujete, a one koje vam zatvaraju put da pogađate laserom. Kada stignete do cilja, automatski prelazite u drugi nivo igre. Sada je pejsaž nešto složeniji, planinski masiv veći, a rakete koje vas napadaju opasnije. Posle drugog sledi treći i četvrti nivo. Ako uspete da dođete do petog, a nalazite na izuzetno složen pejsaž koji ne možete da savladate ukoliko, na osnovu bonusa poena, vaš avion ne dobija neke nove komande.

„Penetrator" poseduje još neke jedinstvene osobine. Ukoliko vam pejsaži dosade ili vam se učine previše složenim (ili previše jednostavnim?), možete da definišete svoje i snimite ih na kasetu. Na taj način igra praktično nikada ne prestaje da biva interesantna, a enormno složeni pejsaži mogu da budu lep poklon za nekog (ne) prijatelja koma ćete pokloniti ovaj program. Penetrator košta 7 funti (kod nas 200 dinara) i mi ga toplo preporučujemo svakom vlasniku Spectruma koji voli akcione igre.

Akcione igre su, sve u svemu, interesantne samo za određeni broj vlasnika računara. Nekoliko igara ovoga tipa svakako treba imati, ali ne previše — samo ćete gubiti vreme i bacati novac.

Računarom po zemlji čuda

Šah uvek možete da igrete i sa prijateljem, akcionu igru u predvorju nekog od bioskopa, ali igru-avanturu ne možete da zamislite ako nemate kućni računar. Igre ovog tipa imaju ogroman broj obožavalaca (u koje spada i autor ovoga teksta), iako je svima jasno da one predstavljaju najgori mogući način za gubljenje vremena i izvor nerviranja koji nikada ne presušuje!

Da bismo vam predstavili kako izgledaju igre-avanture, ispričaćemo skraćeni scenario jedne od njih. Zli čarobnjak je došao u vašu zemlju, isterao dobrog kralja iz njegovog dvorca i kidnapovao njegovu jedinu kćerku-princezu. Zahvaljujući svojoj ogromnoj čarobnoj moći, on je začarao čitav višespratni dvorec i popeo se na njegov krov na kome kuva konačnu mađiju. Vaš zadatak je da uđete u dvorac i, ispitujući ga, pronađete u njemu tragove (naravno, nemate pojma kakvi ti tragovi treba da budu) na osnovu kojih ćete zaključiti gde se nalazi princeza. Kada oslobodite princezu, treba da se popnete na krov i ubijete čarobnjaka, da se vratite u prizemlje i izađete pre nego što mađija počne da deluje. Kada vas niko ne bi ometao u



Jurnjava po lavirintu: „Hungry Horace" (Gladni Horacije) predstavlja jednu od najpopularnijih igara za Spectrum

izvršavanju zadatka, ne bi bilo prevelikih problema. No, zamak vrvi od demona, duhova, trolova i drugih neprijateljski raspoloženih kreatura sa kojima treba da se borite da biste ostvarili neke ciljeve. Na početku igre imate neki bonus životnih poena, ako potrošite životne poene, umirete i gubite igru. Za svaku pobedu nad nekom kreaturom dobijate blago koje je ona čuvala. Tako dobijen novac možete da iskoristite da biste kupili nešto što vam je potrebno ili podmitili duha i nešto saznali od njega.

Kako izvršavate zadatak? Najčešće u toku igre morate da kucate naredbe koje računar izvršava. Te naredbe se kucaju kao kratke rečenice na engleskom. ukoliko, na primer, želite da idete na sever, otkucaćete GO NORTH i računar će vam reći na šta ste naišli. Da bi igra bila namoguće teška, na početku vam nisu poznate sve naredbe koje možete da koristite — morate da ih otkrijete da biste dobili partiju. U jednoj igri, na primer, pauk je čuvao blago i stepenište za sledeći sprat. Uobičajene naredbe za borbu nisu mogle da navedu računar da ubije pauka. trebalo se setiti i otkucati TRAMPLE SPIDER (zgazi pauka) i ovaj demon bi smesta stradao. Dok se ne setite naredbe TRAMPLE, pauk će vas neprekidno trovati i moraćete da započinjete igru od samog početka!

Da bi se rešili problemi koje postavlja neka igra-avanture, treba ispoljiti zavidan talenat za logičko mišljenje i povezivanje činjenica. No ni ovakav talenat često nije dovoljan — ponekad poginete na neki sasvim glup način koji nikakvom logikom niste mogli da predvidite. U igri u kojoj je figurisao gore pomenuti pauk koga treba zgaziti, igrač koji je, posle desetak časova neprekidnog gledanja u ekran, stigao na poslednji sprat biva obavešten da ispred sebe vidi prozor. Sasvim prirodna je igračeva želja da pogleda kroz prozor. No, čim otkuca tu naredbu, računar ga obaveštava da je do njega dopro čarobnjakov otrovni dah (čarobnjak se nalazi na krovu baš iznad tog prozora) i da je nastradao. Igra počinje iz početka i potrebno je novih 5-6 časova da se stigne do poslednjeg sprata!

Ako kupite igru-avanturu i u uputstvu pročitate šta je njen cilj, smatrajte da ste dobro prošli. Ima i igara u kojima ne znete šta vam je cilj! Na osnovu tragova na koje nailazite treba da odredite šta treba da se uradi pa da zatim to i uradite. Ovo nije nemoguće, ali zahteva utrošak značajnog vremena. Nije nikakvo čudo da igrač frustriran nekom igrom ovakvog tipa poželi da analizira program i tako pronađe rešenja zagonetki koje ga muče. Ako je program pisan na bejziku, ovaj zadatak se može izvršiti i za njega će biti potrebno daleko manje vremena nego za pronalaženje rešenja kroz igru, ali ukoliko je program pisan na mašinskom jeziku — ne pokušavajte da „varate". Ima vlasnika računara kojima je mnogo veće zadovoljstvo da analiziraju program i otkriju njegove tajne nego da to učine na „pošten" način. Ne treba se ljutiti na njih — programe ovoga tipa kupujemo da bismo se zabavili, a svako se zabavlja na svoj način.

Od nemoćnog bića do fudbalskog menadžera

Za Spectrum je napisano mnoro igara-avantura. Prema nekim rejtinzima iz stranih časopisa, najefektnija od njih je igra „Hobbit". U njoj igrač predstavlje nemoćno



biće koja treba da prođe kroz svet divova, damona i trolove, ukrada blago koje čuva zmaj i smesti ga u kutiju. Da uspeta u toma morate da se sprijateljite sa stvoranjima koja žive u tom svetu i da računate na njihovu pomoć kada se nadata u nevolji. Ova igra vam na daja vremana da razmišljata previše. ukoliko počnete da sa premišljata o sledećem potazu, igra počinja da se razvija sama od seba, a to vam sigurno neće doneti ništa povoljno. Hobbit je neobično skupa igra, košta 15 funti, ali zahvaljujući preprodavcima stranih programa do nje može da se dođe za svega 300—500 dinara.

Spectres (Utvare) ja još jedna igra-avantura pisana, naravno, za Spectruma. U njoj ja zadatak da se osvetli kuća-lavirint u kojoj se nalazi nekoliko duhova. U tom cilju igrač traba da postavi nakoliko sijalice i poveže ih sa generatorima. Duhovi baža od svetla i osvetljeni nisu opasni. No, ako u mraku naletita na duha — u nevolji sta. Ovaj program košta 11 funti ili, u Jugoslaviji, 300 dinara.

Igre-avantura nisu uvek previše složane ima i simpatičnih igrica koja ne izazivaju previše narviranja, a jedna od najpoznatijih ja „Galaxy". Ovo ja stara igra koju su razvili studenti američkog MIT univarzitata. Igra ja inspirisana serijom „Zvezdava staze" koja se prikazivala i kod nas, iako ja se neki mladi čitaoci ovih redova varovatno ne sećaju. Kao komandant svemirskog broda „Enterpriza" traba da pronađete i uništite klingonske vasionska brodove koji unose nared u vasionu. Čitave vasiona je podeljana na određen broj kvedranata, a svaki od njih na 100 sektora. Na raspolaganju su vam lasari, fotonska torpeda, skaneri kratkog i dugog domata i sistami za pokratanja broda. Njihovom pažljivom upotrebom možate da pratražita vasionu i pronadata neprijateljske brodove, pa i da uništita neke od njih. Pri tom će vaš brod biti oštećen i potrošićeta zalihe energija, pa ćeta morati da obidete neku od vaših baza (i baza treba pronaći) i da se tamo odmorite. Igra ovog tipa ja daleko jednostavnija od svih prethodnih ali ne i manja zanimljiva.

U nekim igrama treba da oprobate diplomatske sposobnosti. Od vas može da se zahteva da vladate vasionom, državom ili da budate običan menadžer nekog fudbalskog kluba. Bilo kako bilo, morate da donosita odluke o kupovinama, prodajama, razmani, ulaganju novca u određene svrhe i slično. Obzirom da često govorimo da ja „svaki Jugosloven fudbalski selektor", vlasnika Spectruma ća verovatno interesovati program nazvan „football Manager" (fudbalski menadžer) koji košta svega 7 funti (kod nas svega 350 dinara). U ovoj igri treba da osvojite angleski kup pažljivo sekupljajući akipu i određujući „pravu jedanaestorku" za svaki meč.

Sve igre-avanture su u izvesnoj meri „potrošno materijalno dobro". Kada prvi put pobedite upoznaćete neka tajne igre i u svakoj sledećoj upotrebi ona će vam biti sve manja zanimljive. Neke od njih igara postaju potpuno neinteresantne posle razumevanja njihovih tajni, dok druge, zahvaljujući ugrađenim generatorima slučajnih brojava, svaki put „zamišljaju" drukčiji lavirint i tako zadržavaju vašu pažnju nekoliko meseci. Igra drugog tipa je verovatno bolje kupovati ne samo zato što daju dužu zabavu nego i zato što su po pravilu jedncatavnija. Još jedan savet: ukoliko kupujete neku igru-avanturu, nija loše da se pobrinete da program buda pisan na bejziku jer tako ostavljate sebi neku vratu odatupnice — ako ne možete da pobedite sve protivnike, možete da analizirate program i pronađete rešenja. Ukoliko kupite mašinski program, možete lako da se nađete u situaciji koja je poznata nekim vlasnicima HP87. Najpre ja korisnička biblioteka za 8 dolara prodavala program „Dungeon of Death", (Kula smrti) a zatim ja jedan od korisnika prodavao uputstvo za rešavanja problema za „samo" 80.

Edukativni programi se, baš kao i u ovom prikazu, povezuju se igrama ali takav atav nije pravilan. u okviru edukativnih programa su skrivene mnoge igre ali oni moraju da pruže daleko više.

Pomoću edukativnih programa najmladi mogu da savladavaju školsko gradivo iz različitih predmeta kroz igru. Najpopularnija je, narevno, matematika koja pred mnogu decu postavlja prateška problema. Deca, na svu sreću, vole računara i nija im teško da provedu časove naporno gledajući u ekran i kucajući tražene odgovore samo da bi izmamili „osmeh" ove intaligentne naprave. Računar je, sa svoje strane, beskrajno strpljiv učitelj: u stanju je da ponavlja potrebna objašnjenja proizvoljan potraban broj puta i da na raznim primerima objašnjava istu stvar sve dok ova datetu ne postane sasvim jasna. Računar je, na žalost, i prilično strog: u memoriju će zapisati sve tačne i netačne odgovora i docnija će, na akranu ili štampaču, dati sumarnu ocenu baz „protekcije".

Nija lako nabaviti ili sastaviti dobar edukativni program. Potrabno je, pored znanja programiranja, i poznavanja dečje psihologija i iskustvo u pedagoškom radu da bi program bio zanimljiv i privlačan za male korisnika. Jedan izvanredan edukativni program za uvežbavanja matematike dobio ja prvu nagradu na našem Konkursu za najboljim domaći program za kućni računar. Program se zove Mato, dalo ja Mladana Sarajčića iz Zagraba i može da ga dobija svaki čitalac ovog specijalnog izdanja uz trošak od 150 dinara (plus prazna kaseta) i navodanja šifra ZXSP92K5001.

Kako nabaviti program

Kada naiđete na raklamu za neki program i odlučite se za njegovu kupovinu, treba da uplatita potrabnu sumu na adresu firma koja ga prodaje. Za to je potrebno da odete u banku i napišete nalog koji će službenici obraditi i poslati ček. U okviru „svrha uplate" nije dovoljno samo da napišete naslov programa koji žalita nego i računar za koji ja pisan — sasvim ja moguće da firma kojoj se obraćete prodaja taj program u nakoliko verzija. Nekoliko nedelja posle uplate dobićete kasetu sa programom — vrlo je verovadno da će cerina biti mala ili uopšte neće ni biti naplaćena.

Krajnja ja vrama da pomanemo jedan izuzetno interesantan klub koji će, uz lakše grupne narudžbe, rešiti i veliki deo problema vezanih za izbor programa. Adresa ovog kluba ja *Micro-Computer Software Club P.O. Box 166, Oxford, OX2 9BJ England*. Dovoljno je da na ovu adresu pošaljete pismo u koma ćete, pored imena, prazimena i adrese navesti koji računar posedujete i sa koliko memorije (Klub za sada okuplja vlasnika računara ZX81, Spectrum BBC, Dragon, VIC, Commodore 64 i Oric) i pridružićete se grupi od preko 30000 članova Kluba. Članarina nema! Dobijaćete redovno neku vrstu časopisa Kluba sa opisima programa koja komisija Kluba proglasi za posebno zanimljive. Da bi se program našao na ovoj listi, on mora da zadovoljava neke prilično stroge uslove, kratko rečeno — da buda vrhunski. Nemate nikakve obaveze da kupujete programa od Kluba ali ako se na to odlučite, imaćete popust od desetak procenata u odnosu na tržišnu cenu programa!

Podaci o MicroComputer Softwara klubu zvuče suviše dobro da bi bili istiniti. I autor ovog teksta je tako mislio sve dok nije probao i uverio se da Klub daje čak i više od onoga što obećava!

PARTNER — pojam savremene i svrsishodne računarske primene

vaš najbolji partner

Prof. dr Anton P Železnikar, RO ISKRA DELTA

Partner je nov mikroračunarski proizvod Iskre Delte za integrisanu obradu podataka u proizvodno-poslovnim i tehničkim primenama i obradi tekstova. Namenjen domaćem, a prvenstveno inostranom tržištu, PARTNER predstavlja plod domaćeg razvoja računarske i programska oprema.

Partner je nastajao u skromnim prilikama, ali sa sluhom za savremenu tehnologiju, za neophodne međunarodne tržišne reference u njegovoj primeni i za njegovu konkurentnu sistemsku sposobnost. Izbor mašinske i programske struktura bio je podvrgnut najoštrijim međunarodnim konkurentnim zahtevima.

Razvoj PARTNERA bio je na samom početku uslovljen brižljivim izborom razvojnih kadrova koji su se smelo i naoružani razvojnom metodologijom upustili u borbu sa mnogim promenljivim zahtevima tržišnih faktora i koji su u sebi našli snagu za beskompromisnu takmičarsku igru u međunarodnom konkurentnom prostoru. Iskra Delta je ovim posebnim radnim pristupom našla put za svoj budući korak u osvajanju proizvodnja i traženju malih računarskih sistema.

PARTNER u zemlji i u svetu

Među raznovrsnim mikroračunarskim sistemima, PARTNER zauzima svoje posebno mesto. U razvijenom svetu se proizvode različiti mikroračunari, kao npr. ručni (cena od 200 do 300 dolara), domaći (od 100 do 600 dolara) i lični računari (raspon do 20000 dolara). Ručni i domači računari upotrebljavaju kasetne trakaste memorije s lični fleksibilna i tvrde diskova na kojima može da se pohrani i do 600 miliona znakova. Cena ličnog računara dostiže svoju najveću vrednost kod pravog višekorisničkog sistema koji omogućuje vremenski nesmetani i istovremeni rad više korisnika.

PARTNER je lični računar za poslovne obrade i ima dva disk jedinice: fleksibilni: (1 milion reči) i tvrdi disk (10 miliona reči) Upotrebljava najnoviji operacioni sistem za osamobitne mikroprocesore (CP/M Plus) Njegova izvedba je stopa, diskovi se nalaze u kućištu sa ostalom elektronikom, kućište nije veće od srednjeg (prenosnog) televizijskog aparata. Tastatura i štampač su dodatne jedinice.

Poslovni programski paketi su prilagođeni pojedinim tržištima domaćem i inostranim (austrijskom i nemačkom) Standardni programski paketi su glavna knjiga, saldakonti kupaca i dobavljača, skladišno poslovanje i fakturisanje, ovi paketi su uključeni u cenu sistema. Ovakav sistem omogućuje odgovarajuće obrade podataka srednje velikim preduzećima (do 500 zaposlenih), tako da više ne moraju da iznajmljuju spoljašne usluge i da povećavaju broj zaposlenih Pored standardnih pakata, na raspolaganju su, takođe drugi paketi, kao npr. obračun ličnih dohodaka.

Bitna primena PARTNERA leži, svakako, i u obradi tekstova, pisanju dopisa, stručne i upravne dokumentacije, originalnih tekstova (književnici, pesnici, novinari, govornici), budući da se tekstovi mogu po želji oblikovati, menjati, popravljati, dopunjavati, preuređivati i pravopisno korigovati Ova kancelarijska i lična funkcija PARTNERA znatno olakšava rad piscima Na jednu disketu možemo da pohranimo tekst od 40 do 50 autorskih tabaka.

Neracionalnost velikih sistema

Zašto u određenim sredinama još uvek težimo upotrebi velikih, skupih računarskih sistema, koje čak ne znamo u celini da upotrebimo, i time ih pretvaramo u proizvođače obimne, nerelevantne dokumentacije. Ideologije velikih sistema doživljava svoj zalazak i u razvijenom svetu, gde se sve više širi upotreba malih računarskih sistema. Tržište velikih sistema stagnira. Jučerašnji proizvođači ovih sistema spremaju se listom za izlazak na tržište. Ovo važi kako za čuveni IBM, tako i za japanske proizvođače.

Preduzeće IMB „šalje u penziju" svoje sisteme iz porodice 4300 koji su se pojavili na tržištu godina 1979. Konsultantske organizacije na Zapadu odgovaraju korisnike od kupovina ovih sistema. Međutim, u zaostalim sredinama u toku su intenzivne pripreme baš za kupovinu tih isluženih sistema, uz jasnu težnju da se istisnu i poslednji dolari iz sekundarnih tržišta. Ovde domaća nesposobnost i velik proizvođač prijateljski stišću ruku

U Jugoslaviji tek treba da oblikujemo tržište mikroračunara. Radi se o oblikovanju kvalitetnijeg i masovnijeg tržište nego što je tržište klasičnih velikih sistema. O podizanju znanje i osposobljenosti korisnika, o novoj vrsti obrazovanja, o mnogo racionalnijoj upotrebi koju veliki sistemi, sa svojom udaljenošću, praktično nisu nikada dotakli. Radi se, dakle, o novom impulsu velikom delu privrede, o podizanju kvaliteta privređivanja i osvajanje novih oblika rada.

Napadi na domaći proizvod

Kod nas su za određeni tip industrije izmislili neukusnu psovku. Savremena, pogotovo tehnološka proizvodnja (elektronska, računarska, kancelarijska, telekomunikaciona, automobilska, mašinska itd) zasniva se na sastavljanju podsistema i sistema iz komponenata. U Japanu, u Nemačkoj, u SAD, novinarima ne pada ni na pamet da ovaj tip industrije pogrdno nazivaju „šrafciger" industrijom, jer sredina ove industrije zna da se radi o veoma komplikovanim, tehnološki složenim proizvodima koji se u većini slučajeva mogu proizvoditi samo uz velike napore (razvojne, proizvodne, tržišne, obrazovne, poslovne).

Zapad je za industriju ove vrste uveo pojam OEM (Original Equipment Manufacturer) Proizvođač koji prodaje drugim proizvođačima svoje originalne podsisteme (periferne računarske jedinice, elektronske module, programske module, operacione sisteme, itd.) jeste OEM, ako prodaje svoje proizvode pod određenim uslovima (količine, cena, kvalitet). Proizvođač koji sastavlja svoje sisteme od OEM delova može da bude OEM proizvođač na višem proizvodnom nivou, a može da bude i konačni proizvođač. Time OEM lanac može da bude veoma dugačak i rasteže se od komponenata, podsistema do sistema i dalje

Proizvodnja mikroračunarskih sistema je, po pravilu, OEM proizvodnja finalnog proizvoda, ali je ovde svakako bitno da sistem kao finalni proizvod nije licencni proizvod. PARTNER nije licencni proizvod i kao računarski sistem predstavlja originalno domaće dostignuće. Zato može i u izvozu da bude dohodovno uspešan



Uspešan i u svetu: PARTNER na Sajmu Cebit Hanover 1983.



Neopravdani napadi na domaću industriju: Dr Anton P. Železnikar

5 hakerska lektira

Specijalizovani časopisi i knjige

Klubovi programera pradatavljaju izvanredan način da aa upoznaju naka osobina računara i da aa dođa do upotrabijivih programa. Drugi izvor informacija su komercijaini čaaopiai. Oni, u izvaanom amialu. pradatavljaju auprotnoat čaaoplaima koja izadaju vači klubovi programara, jar ja zarada imparativ izdavačkih kuča. Zato au komarcijaini čaaopiai prapuni blaštavih raklama u kojima avaki proizvođač hvali svoj kompjuter, a autor avoj proizvod. U njima mogu da aa nađu i vaoma intarasantni napiai koji aa fanomanološka atrana tratiraju bum računara, portreti alaktroničara, programara i proaačnih korianika računara, kao i ralativno dobra škola programiranja. Međutim, ako u nakom komarcijalnom čaaopisu žalite da nađate dobar program, to će vam taško poći za rukom. Raziog ja iako pogoditi: vačina dobrih programa za atona računara ja komarcijalizovana i prodaja aa na kaaatama; ratko ja ko apraman da objavi program u čaaopiau i tako sa iiši primamljiva zarada.

U komercijalnim časopisima, ipak ima programa, ali su to mahom jednostavne rutine iz kojih se, doduše, uči programiranje i upoznaju neke osobine pojedinih računara ali koje ne pružaju praviše časove zabave Tekstove koji otkrivaju manje poznate osobine računara treba ‚ražiti svećom Moglo bi se zaključiti da pretplata na neki komercijalni časopis nema nikakve svrhe No, nije tako U komercijalnim časopisima mogu da se nađu napisi koji upoređuju neke hardverske dodatka i programe po vrednosti i tako vam pomažu da nabavite one koji su vam potrebni Vrednost komercijalnih časopisa najbolje možemo da ilustrujemo situacijom u kojoj smo se mi nalazili pri pripremanju ovog specijalnog izdanja Pomenuli smo desetine računara, isto toliko hardverskih dodataka i priličan broj programa Jasno je da nismo bili u prilici da sve njih stavimo na neki od stolova u redakciji Ipak, konsultacija priličné gomila časopisa nam je pomogla da shvatimo dobre i loše strane svakog proizvoda i da pokušamo da ih realno prikažemo U sličnoj situaciji ćete se i vi, verovetno, naći kada poželite da kupita neki program U reklami (koju, naravno, nalazite u nekom komercijalnom časopisu) možete da pročitate samo nekoliko reči o svakom programu Međutim, pogled na prikaze programa će vam otkriti da je neka igra, na primer, neobično dinamična ali da ne zahteva neko posebno razmišljanje, da je druga toliko teška da če vam oduzeti tri meseca, dok je treća u svakom smislu preporučljiva, ali da nema naročite zvučne efekte

„Your Computer"

Čini nam se, ipak, da nema smisla pretplaćivati se na više od jednog komercijalnog časopisa U većini ćete naći potpuno iste reklame koje se ponavljaju iz broja u broj, na slične programe i njihove prikaze Većina jugoslovenskih pretplatnika se, bar prema našim iskustvima, odlučuja za časopis „Your Computer" To je, inače, najtiražniji britanski časopis posvećen isključivo stonim računarima Najveći deo prostora je do skora bio posvećen Sinclairovim računarima, ali se ovaj odnos u poslednje vreme pomera ka računaru Commodore 64, BBC modelima i Oric-u Pored prigodnih vesti i napisa, časopis održava posebnu rubriku koja je posvećena programima Prilozi za ovu rubriku su obično prosečnog kvaliteta, iako tu i tamo blesne po neka interesantna rutina Časopis je izvanredno tehnički uređen, izlazi 12 puta godišnje i, za pretplatnike van Engleska, košta 18 funti godišnje Ako želite da se pratplatite, obratite sa na adresu **IPC Buainesa Presa Oakilft House, Perrymount Road, Haywarda Haath, Suasex RH16, England 3RD.** ICC Business Press izdaje i časopis „Practical Computing" koji ima dužu tradiciju i niži tiraž od „Your Computera" Ovaj časopis je orijentisan na nešto skuplje računare, pa mu je varovatno zato i pretplata viša 22 funte godišnje Časopis je prilično kvalitetan i izgleda da u njemu ima za nijansu manje reklama nego obično Posebno bismo ga preporučili vlasnicima BBC modela računara DAI i drugih mašina iz „više srednje" klasa

„Sinclair User"

Vlasnicima Sinclairovih računara na raspolaganju je časopis „Sinclair User" Časopis izlazi jednom mesečno u izdanju **ECC Publicationa, 196—200 Balla Pond, London N1 4AQ.** Tehnički kvalitet i sadržina časopisa zaostaju za „Your Computer"-om i „Practical Computer"-om ali časopis ima dobru stranu da vlasnik ZX81 i ZX Spectruma ne mora da plaća za informacije o drugim računarima koje ga ne interesuju Formalno posmatrano, „Cinclair User" je nezavisan od firma Sinclair Research, ali se njagovi urednici očigledno dobro čuvaju sukoba sa „ujka Klivom" — svaka kritika Sinclairovih modela je „zamotana u devet oblandi" i sakrivena u nekom dužem tekstu U časopisu izdvajamo jednu rubriku „Helpline" koju uređuje Endrju Njuson (Andrew Hewson) Iz meseca u mesec Endrju na ove dve-tri strane odgovera na najrazličitija pitanja vlasnika ZX računara sa mnogo duha i, što je još važnije, znanja Čak i iskusan vlasnik ZX računara će neučiti mnogo iz ovakvih napisa, pošto če mu biti skrenuta pažnja na neke pojedinosti koje mu obično izmiču

Pošto je broj vlasnika Sinclairovih modela poslednjih meseci porastao pojavila su se još dva časopisa koja izdaje ista kuća Prvi je namenjen onima koji žele da dođu do programa i zove se, naravno, „Sinclair Programs" Časopis „Sinclair Projacts" je namenjen onima koji žele da sastavljaju hardverske dodatke koji će povećati sposobnosti njihovih računara Sa časopisom „Sinclair Projects" imamo relativno malo iskustva dok časopis „Sinclair Programs" teško možemo da preporučimo — čini nam se da su programi koji se tamo objavljuju na granici trivijalnosti

„Computing Monthly" i „Byte"

Eminentni časopis „Elactronics" takođe donosi određen broj samogradnji koje su vezane za računare Ovaj časopis je ipak namenjen poglavito elektroničerima i onima koji se ovom oblašću vrlo ozbiljno bave — tekstovi podrazumevaju nivo znanja koji „laici" ne poseduju Časopis „Computing Monthly" je namenjen onima koji se ozbiljno bave programiranjem — u njemu nećete naći napise posvećene početnicima i objašnjenja osnovnih pojmova Onaj ko ima strpljenja i znanja da pređe često suvoparne tekstove koja objavljuje ovaj časopis može da očekuje da će se njegovo poznavanje nauke koja se zove programiranje unapređivati iz meseca u mesec Sličnu ali ipak nešto popularniju filozofiju prati još eminentniji časopis „Byte" koji, doduše ne možemo da preporučimo početnicima ali koji može da bude koristan za sve one koji su u programiranju napravili prve samostalne korake

„Chip"

Svi pomenuti časopisi izlaze na engleskom jeziku Na treba pomisliti da je potrabno izvanredno poznavanje ovog jezika da bi se neki napis pročitao — svi autori se

truđe da pišu što popularnijim stilom. Čitaoci kojima je nemački bliži opredeliće se za časopis „CHIP". Reč je o tehnički izvanrednom i inventivno ilustrovanom časopisu u kome ćete, pored prikaza računara i programa, naći uputstva za samogradnju, napise o razvoju računara i druge vesti. Sve u svemu, časopis vredi i više od 85 maraka godišnje koliko iznosi pretplata.

„A & B Computing"

Vlasnicima BBC mikrokompjutera možemo da preporučimo časopis „A & B Computing" i neke francuske časopise. Francuski časopisi imaju poseban šarm i autoru ovog teksta može samo da bude žao što mu nepoznavanje jezika stvara ogromne teškoće pri njihovom praćenju. Jedan od najpoznatijih je „l'Ordinateur individuel" koji tretira mnoge računare, uključujući i Sinclairove modele, Commodore, BBC mikroračunare, TRS 80 i džepne HP41C, TI59, PC1211 i PC1500. Za vlasnike džepnih računara je pokrenut časopis „l'Ordinateur de Poche". Njegov kvalitet je na prilično niskim granama, ali se postepeno poboljšava. Oba časopisa mogu da budu posebno interesantna za vlasnika PC1500 pošto objavljuju veoma zanimljive napise o programiranju na mašinskom jeziku, koji je do skora bio jednačina sa mnogo nepoznatih.

„MC microcomputer"

I italijani izdaju nekoliko neobično interesantnih časopisa, izdvaja se „MC Microcomputer" u izdanju Officio Abbonamenti. Časopis je interesantan kako za vlasnika HP41C i TI59 tako i za one koji poseduju veće Hewlett-Packardove modele, Acornova računare i Newbrain. Podrška Sinclairovim modelima je veoma umerena, pre svega zato što časopis pošteno izlaže svoju koncepciju — obzirom da su programi za stone modele komercijalizovani, MC Microcomputer ne želi da objavljuje rutine koje su tako loše da niko neće da ih otkupi. Časopis je, dakle, pun interesantnih napisa o računarima i prikazima novih modela, iako ponekad zaluta i neki interesantan program za TRS 80.

U čitavom ovom prikazu izostavili smo američke časopise i to sa razlogom: američko tržište se u mnogome razlikuje od evropskog, pre svega po tome što ne postoji interesovanje za jeftinije računare kao što su Sinclairovi modeli (u SAD se ZX81 i Spectrum prodaju u nešto izmenjenim verzijama pod imenom Timex-Sinclair). Najveće je interesovanje za IBM PC i HP 87 — računare koji su za naš džep preskupi. Navodimo, ipak, adresu časopisa koji je posvećen Timex-Sinclairovim modelima obzirom da nam je iz iskustva poznato da u Jugoslaviji postoji određen broj zainteresovanih: **Timex Sinclair User, ECC Publications, 196—200 Balls Pond Road, London N1 4AQ.** Primećujete da je adresa engleska: ukoliko pišete njima proći ćete znatno jeftinije nego da časopis naručujete od američkog izdavača — 20 funti godišnje.

Računarske početnice

Za razliku od časopisa koji predstavljaju potrošnu vrednost", knjiga je nešto što treba da vam pomogne da bolje savladate i racionalnije koristite računar. Neke od njih, kao što ćemo videti, predstavljaju samo uvod u čarobni svet računara, druge vas uče bejziku i mašinskom jeziku, a treće su namenjene onima koji žele da se bave teorijom algoritama i velikim kompjuterskim sistemima.

Ako ste tek dobili ZX81, možemo da vam preporučimo knjigu „Getting Acquainted With Your ZX81" koju je napisao Tim Hartnal (Hartnell). U njoj nećete naći mnogo teksta — dominiraju programi koje možete jednostavno da otkucate i tako demonstrirate sebi i prijateljima mogućnosti vašeg novog računara. Odgovarajuća knjiga posvećena vlasnicima Spectruma je delo Jana Stjuarta (Ian Stewart) i Robina Jonesa i nosi naslov „Computer Puzzles for Spectrum and ZX81" (iako figuriše u naslovu, ZX81 je malo zastupljen).

Obe knjige su namenjene onima koji se nalaze na samom početku. No, iz njih ne može mnogo da se nauči o programiranju — objavljeni programi su toliko složeni da je početniku onemogućeno bilo kakvo razumevanje. Za one koji prestanu da se igraju i požele da napišu neki samostalni program (a ta sudbina obično zadesi svakog vlasnika računara), napisane su mnogobrojne „početnice" — knjige koje na popularan način objašnjavaju osnovne programerske tehnike. Ovakvih knjiga na tržištu ima dosta, a cene su takve da ste u prilici da ih nabavite više od jedne. No, ne treba da očekujete da će gomila knjiga poređanih na vašem stolu predstavljati sigurnu ulaznicu u svet programera. Površno čitanje knjiga vam neće pomoći da napišete prve programe — programiranje je nauka a ne algoritam koji se jednostavno memoriše. Zato se zadovoljite jednom dobrom knjigom — početnicom i pokušajte da je savladate isprobavajući svaki program na računaru i menjajući neke njegove delove radi provere razumevanja materije.

Jedna od odličnih početnica nosi naslov „Introduction to Computer Programming" i delo je Brajana Smita (Brian Reffin Smith). Knjiga je obilno korišćena pri realizovanju „Osnovne škole bejzika" koju smo objavili u ovom specijalnom izdanju, pa verovatno neće biti od interesa za čitaoce ovih redova. Istovremeno dobra i loša strana ove knjige je što je namenjena vlasnicima svih stonih računara, za vlasnike ZX81 je napisano nekoliko specijalizovanih udžbenika koje možemo da vam preporučimo. To je najpre knjiga Randla Harlija (Randle Hurley) „ZX81 — Programming for Real Applications". Za razliku od većine sličnih, knjiga se ne zanosi „edukativnim" programima koji počinju od toga da čitaocu predlože da otkuca PRINT „ZX81" i začudi se rezultatu koji se pojavio na ekranu. Umesto toga, obrađeno je nekoliko korisnih problema (počevši od tekst-procesora) i prikazan put kojim je tekao njihov razvoj. Vlasnici Spectruma mogu da nabave knjigu „Talk to Your Spectrum" koju izdaje kuća Melbourne Publishers iz Londona.

Tajanstveni svet mašinskog jezika

Ista kuća izdaje i neobično interesantnu knjigu „Spectrum Machine Language for Absolute Beginner", čijim pominjanjem počinjemo paragraf posvećen onima koji su ovladali tajnama bejzika i žele da se upuste u tajanstveni svet mašinskog jezika.

Knjiga počinje od objašnjenja na nivou deteta od 10 godina, ali se već u drugom delu pojavljuju i neki prilično složeni problemi koji mogu da dovedu do očajanja onoga ko, pre čitanja ovog poglavlja, nije „ispekao zanat" na bejzik programima i shvatio šta je to algoritam i kako se uopšte neki problem algoritamski rešava. Za one koji imaju malo više poverenja u svoja sposobnosti i znanje, preporučujemo knjigu „Mastering Machine Code on your ZX81" (knjiga je vrlo korisna i vlasnicima Spectruma) koju je napisao Toni Barker. Interesantna je i knjiga Stivena Hantera (Stevena Huntera) „Make it Microcode", delo pisano na prilično avangardan način, uz mnogo šala i anegdota. Poslednja knjiga pruža nekoliko prijatnih časova, ali se iz nje teško može naučiti nešto osim ponekog „kompjuterskog vica".

Sve pomenute knjige su namenjene onima koji žele da se upoznaju sa bejzikom ili mašinskim jezikom. No, i vlasnici računara koji već poznaju ove oblasti mogu da požele da nabave knjigu koja će im otkriti tajne koje krije njihov kompjuter. Postoji određen broj vlasnika ZX81 i Spectruma koji su već savladali mašinski jezik procesora Z80 radeći na nekom drugom računaru. No, čak i pravim majstorima preporučujemo knjigu „The Complete Spectrum ROM Disassembled" koju su napisali Dr Jan (Ian Logan) i dr Frenk (Frank) O'Hara. U ovoj knjizi možemo da pronađemo početne adrese svih važnijih rutina u ROM-u Spectruma koje mogu da se koriste u mašinskim programima korisnika. Zašto, na primer, da sami sastavljamo program koji množi dva decimalna broja, ako se optimalno rešenje ovog problema već nalazi negde u ROM-u? Poznavanje ove knjige omogućiće da mašinski programi koje pišemo budu maksimalno racionalni i pregledni.

Još jedna knjiga ovog tipa ima primamljiv naslov „40 Best Machine Code Routines for the ZX Spectrum" i delo je Johna (John) Hardmana i Endrjua Njusona. U izvanredno napisanom uvodu izložene su osnove mašinskog jezika koje su onome ko poznaje programiranja dovoljne da u potpunosti upozna ovaj procesor. Slede 40 mašinskih programa, najveći deo je, kao što se moglo pretpostaviti, sasvim beskoristan, ali tu i tamo naiđemo na poneki potprogram koji je vredan novca utrošenog na ovu knjigu. Kao i većina spomenutih naslova, knjiga može da se dobije od **ZX81 & Spectrum Software Books, Cambridge Computer Store, I Emmanuel Street, Cambridge CB1 1NE.**

Onima koji se razumeju u elektroniku i žele da prošire mogućnosti svog računara za sada stoji na raspolaganju relativno malo dobre literature. Spomenućemo „ZX81 Add-on Book", kao i „20 Simple Electronic Projects for the ZX81 and other Computers" od Stivena Adamsa (Stephen Adams). U obe knjige se polazi od jednostavnih projekata sumnjive upotrebljivosti i dolazi do složenih dodataka koje će sa retko koji vlasnik računara rešiti da napravi. Sve u svemu, oni koji žele da proširuju računar upućeni su uglavnom na stručnu literaturu elektroničara.

Umetnost i nauka programiranja

Postoje, najzad, i knjige koje tretiraju programiranje kao nauku. Ima ih zaista mnogo i teško je pomenuti one koje se izdvajaju toliko da bismo ih mirne savesti

Evropski veteran· Specijalno izdanje časopisa „Personal Computer World", prvi te vrste u Evropi, koji je osnovao i jedno vreme vodio Jugosloven Anđelko Zgorelec, časopis izlazi i danas kao najobimniji britanski magazin za kućne računare

Sve više reklama — sve manje tekstova Popularni „Sinclair User"

Poplava računarska lektira: Jedan od naslova objavljenih u Nemačkoj

preporučili čitaocima Čitanjem neke od ovih knjiga nećete naučiti ništa što će biti trenutno primenljivo, ali će studiranje njihovog sadržaja unaprediti vaše programerske sposobnosti i otvoriti vam nove vidike u ovoj oblasti. Neka od knjiga ovog tipa su pisane sa ambicijom da bar u izvesnoj meri budu popularne (jedna od takvih je čuveno delo S H. Holingdejle (Hollingdale). i G C. Tuslla (Tootill) i nosi jednostavan naslov „Electronic Computers") dok su druge namenjena profesionalcima (pomenimo samo čuvenu monografiju Donalda Knutha „The Art of Computer Programming" u više tomova) Ovde ćemo, prema abecednom redu imena autora, navesti nekoliko naslova koji su nam bili na raspolaganju u toku pisanja ovog specijalnog izdanja, a koji bi mogli da budu od interesa za one koji žele da nastave neka naša proučavanja, ali na višem nivou·

• Martin J & Norman A. R. D. „The Computerized Society", Penguin Books 1979

• Michie D: „On Machine Intelligence" Edinburgh University Press, 1974

• Rowe B C. „Computers and You", National Computing Centre 1972.

„A Scientific American Book", W. H. Freeman 1966 (bilo je i dosta docnijih izdanja)

U ovom specijalnom izdanju korišćeni su i materijali iz časopisa „Your Computer", „What Micro", „Sinclair User", „Practical Computing", „L'Ordinateur Individuel", „Byte", i drugih. Od domaće literature od koristi nam je bila knjiga Vojislava Stojkovića i Borivoja Stankovića „Uvod u programiranje", kao i delo Dragana M Pantića „Džepni kalkulator", koje, iako danas prevaziđeno (izdato je još 1976), predstavlja jedan od pionirskih poduhvata na našem tlu Domaća literatura iz oblasti o kojima pišemo se, naime, do skora svodila isključivo na univerzitetska udžbenika, ali se trendovi polako menjaju — ovo izdanje bi trebalo da podstakne i druge autore da približe računare i programiranje širim krugovima korisnika.

6/ računar „galaksija"

Napravi i ti

Piše: Voja Antonić

Kada kupujate komercijalni model kompjutera, nalazite se u prilično prijatnoj poziciji: dobijate računar koji se jednostavno povezuje sa televizorom, uključujete ga u mrežu i počinjate sa radom. Kada sami pravite računar, stvari su mnogo složenije: počinjete od neugledne gomile čipova, kondenzatora, otpornika i tastera i pokušavate da ih povežete u celinu koja će raditi. Ako budete pažljivo sledili uputstva koja ćemo dati, to ne bi trebalo da bude previše teško: nikakvo posebno predznanje, osim određenog iskustva sa lemljenjem, nije potrebno da bi se, posle određenog broja časova rada, na ekranu vašeg televizora pojavilo ohrabrujuće READY.

Samogradnja, ipak, sa sobom nosi i određene rizike. Iako je računar „galaksija" izrađen već u osam primeraka koji besprekorno funkcionišu, pa nama dilema o ispravnosti samog projekta, postoji mogućnost da naiđete na neki neispravan čip, da nešto pogrešno postavite, napravite „mikronski" kratak spoj na štampanom kolu ili stradate na nekoj drugoj „krivini". Tada će biti neophodno nešto više znanja ili (u ekstremnom slučaju) stručna pomoć da bi se neispravnost pronašla i otklonila. Ipak, stara poslovica kaže da bez rizika nema ni uspeha. Računar koji ste napravili sopstvenim rukama i koji je proradio pružiće vam više zadovoljstva od nekog komercijalnog modela, ne računajući činjenicu da ćete u toku gradnje videti iz čega se kompjuter sastoji i pronići bar u neke tajne njegovog funkcionisanja.

Pre nego što donesete odluku o samogradnji računara, treba da razumete njegovu osnovnu koncepciju. Računar „galaksija" je konstruisan tako da bude pogodan za samogradnju, da je jednostavan i zgodan za rukovanje a ipak dovoljno moćan da može da se upotrebi i u korisnim poslovima i, razume se, za zabavu. U tom smislu, „galaksija", doduše, ne može da konkuriše moćnijim komercijalnim modelima stonih računara, kao što su Apple II, Commodore 64 ili čak ZX Spectrum, ali ne treba zaboraviti da je jeftinija od svakog od njih i da se, što je još važnije, sastoji od delova koji mogu da se nabave kod nas ili legalno uvezu. Čini nam se da je „galaksija" po vrednosti iznad poznatog ZX81, koji je prodat u ogromnom broju primeraka, pokazavši, tako, da je za početak najpodesniji skroman i jeftin računar. Ne treba zaboraviti da, prema rezultatima preliminarnog ispitivanja koja smo izvršili, računar „galaksija" namerava da napravi preko hiljadu naših čitalaca što znači da će uslediti razmena programa, informacija, kao i eventualno osnivanje klubova. Iz iskustva znamo da saradnja vlasnika različitih kompjutera (pa ma koliko ti kompjuteri bili moćni) ne može da bude naročito interesantna i plodna pa će računar „galaksija" možda poslužiti kao most između ljubitelja računara širom Jugoslavije.



Moćniji nego što izgleda. Računar „Galaksija" predstavlja idealnu mašinu za učenje programiranja, ozbiljne primene u svakodnevnom životu i, svakako, zabavu

Hardverska . . .

Jednoslojno štampano kolo

Računar „galaksija" se sastoji od štampanog kola, devetnaest ili dvadeset integrisanih kola (zavisno da li se odlučite da stavite treći čip 6116 i tako proširite RAM memoriju „galaksije" na 6 kilobajta), određenog broja tranzistora, dioda, otpornika i kondenzatora (precizni popis opreme dajemo nešto docnije), tastature, kutije i ispravljača. Štampano kolo je jednoslojno, što znači da je njegova izrada pojednostavljena, da ima manje mogućnosti za greška i da je, na kraju, smeštanje elemenata povezano sa manjim rizikom. Jedna od posledica je da smo morali da upotrebimo priličan broj kratkospojnika („džampera"), što ne bi trebalo da izazova nikakve probleme osim izvesnog produženja posla.

Softverski video

Jednoslojna štampa onemogućava postavljanje previše integrisanih kola (čipova). Želeli smo, uz to, da računar „galaksija" bude što jeftiniji, pa smo morali da težimo da se što više stvari reši programski (softverski), a da broj čipova bude minimalan. Najveće pojednostavljenje hardvera je postignuto time što je video podržan softverski: mikroprocesor, naime, posle svake dve stotinke sekunde prekida redovan posao i pomaže video-stepenu da ispiše sliku na ekranu.

Statičke memorije

Da bi se shema računara maksimalno pojednostavila, upotrebljene su statičke umesto danas prilično popularnih dinamičkih memorija (dinamičke memorije se, okvirno govoreći, sastoje od kondenzatora koje računar mora da „dopunjava" u kratkim vremenskim intervalima). Osim toga, za korišćenje statičkih memorija nije potrebno dovoditi tri stabilisana napona napajanja, što u mnogome pojednostavljuje štampano kolo i, što je još važnije, značajno smanjuje verovatnoću da pogrešite pri izradi (vrlo je verovatno da računar sa dinamičkim memorijama ne bi proradio čak ni kod umereno iskusnog konstruktora). Statičke memorije su, uz to, skoro kompatibilne sa EPROM-om 2716, što znači da se deo RAM-a lako može zameniti ROM-om. Statičke memorije, na žalost, imaju i neke loše strane: skuplje su i kapacitet im je manji, ali nam se čini da je njihovom upotrebom dobijeno daleko više nego što je izgubljeno. Obzirom da RAM memorija računara „galaksija" može da se povećava do 54 kilobajta, iskusnijim konstruktorima je ostavljena mogućnost da upotrebe dinamičke memorije za to proširenja i tako uštede novac.

Standardni čipovi

Jedan od važnih koncepcijskih zahteva bio je i da se koriste standardni čipovi koji

Zašto graditi računar kada se na tržištu nalaze modeli velikih mogućnosti i prihvatljivih cena? Razlog nije teško pogoditi: jednom nerazumljivom zabludom nadležnih organa uvoz računara u Jugoslaviju je, osim kada se radi o povratnicima iz inostranstva, zabranjen. To znači da ljubitelj računara nema legalne mogućnosti da dođe čak i do najjednostavnije mašine i dalje usavršava svoje sposobnosti čak i ako je spreman da plati visoka carina. Kod nas, doduše, postoji nekoliko firmi koje se bave sklapanjem stranih kompjutera i, u poslednje vreme, započinju proizvodnju originalnih modela, ali se cene tih računara, na žalost, kreću između 100 i 900 hiljada dinara (govorimo, razume se, o stonim računarima) pa su teško prihvatljive za džepove pojedinaca. Zato je računar „galaksija", kome su posvećene sledeće stranice, prvi originalni domaći lični računar pristupačan svakome ko želi da uloži trud u njegovo sklapanje!



Praktičan, pouzdan, jeftin: Prvi originalni domaći kućni računar pristupačan svakome

TEHNICKE KARAKTERISTIKE

MIKROPROCESOR	Z80A
ROM	4-8 K
RAM	STATICKI, 2-6 K
MOGUCNOST PROSIRENJA RAM-A	DO 54 K
PROGRAMSKI JEZIK	BASIC INTERPRETER
FORMAT EKRANA	16 REDOVA PO 32 ZNAKA
GRAFIKA	48 X 64 TACKE
ARITMETIKA	POKRETNI ZAREZ,32-BITNA TACNOST
BRZINA SNIMANJA NA KASETU	280 BAUDA
POTROSNJA STRUJE	5 W
DIREKTNI PRKLJUCCI	TV MONITOR
	TV PRIJEMNIK
	KASETNI MAGNETOFON
PROSIRENJA	PREKO STANDARDNOG 44-POLNOG KONEKTORA

se lako nabavljaju. Razlog nije teško pogoditi: zavisni smo od uvoza integrisanih kola, pa svi nestandardni čipovi predstavljaju priličan problem pri narudžbi (službenici naših banaka vas neće oduševljeno dočekati ako poželite da na deset adresa pošaljete po dve tri funte!). Zato nije mogao da bude upotrebljen neki od standardnih generatora karaktera ni keyboard-encoder, pa smo pripremili jedan EPROM od dva kilobajta kome mikroprocesor može da se obrati da bi saznao kako koje slovo izgleda. Osim toga, u ovaj EPROM su mogla da budu upisane i naša latinična slova Č, Ć, Ž i Š, koja omogućavaju da domaći računar bude zaista domaći.

Ograničena rezolucija

Relativno mala rezolucija (32 slova ili 64 grafičke tačke u svakom redu) posledica je toga što mikroprocesor jednostavno nema vremena da ispiše više znakova. Većini čitalaca koji su nam poslali preliminarnu narudžbenicu ova karakteristika je jedino i zasmetala (uz malu memoriju ali je taj problem otklonjen). Moramo da kažemo da se rezolucija računara „galaksije" ne može poboljšati nikakvim hardverskim ni softverskim dodacima koji ne bi promenili strukturu čitavog računara (voleli bismo, naravno, da neki elektroničar ili programer u praksi demantuje ovu izjavu) i da se sa njom svi oni koji odluče da naprave računar moraju pomiriti. Ipak, ne čini nam se da ovo predstavlja preveliko ograničenje: računar „galaksija", to stalno treba imati u vidu, nije namenjen krupnim komercijalnim primenama, koje bi zahtevale obradu teksta, a što se igara tiče, vlasnici ZX81, koji imaju istu rezoluciju, uverili su nas da je za dobrog programera i ona dovoljna da napravi prilično efektne slike.

Profesionalna tastatura

Želeli smo da računar „galaksija" bude opremljen profesionalnom tastaturom iako je njegova cena na ovaj način nešto povećana (mada stoji i pitanje gde kod nas mogu da se nabave ikakve tastature — bolje ili gore od one koju predlažemo) čini nam se da se ova investicija višestruko isplati: u toku kucanja programa, pa čak i u toku najjednostavnijih igara, loša tastatura može da priredi vlasniku računara toliko problema da požali da digne ruke od svog kompjutera zanemarujući sve njegove dobre karakteristike. Dobra tastatura, dakle, predstavlja neku vrstu stimulanse za kreativan rad.

Televizor i monitor

Mogućnosti povezivanja sa dodacima su značajne. Pre svega, računar „galaksija" postaje sasvim beskoristan ako nije povezan sa televizorom. Umesto televizora, može da se upotrebi i monitor (prema propisima koji važe već nekoliko godina svaki televizor nekog od naših proizvođača bi trebao da ima i monitorski ulaz, ali se toga niko ne drži) koji unekoliko poboljšava kvalitet slike, mada je ona uvek dovoljno stabilna.

Rad sa kasetofonom

„Galaksija" se povezuje sa kasetofonom uz ostvarivanje brzine prenosa od 280 bauda (u svakoj sekundi se na traku upiše 280 bita). Ova brzina garantuje pouzdanost samim tim što je manja od one koju nude neki komercijalni modeli. Kao ilustracija problema pouzdanost/brzina neka posluži citat iz uputstva za upotrebu BBC računara koji se reklamira kao jedan od najkompletnijih i najjačih stonih računara na tržištu: „brzina snimanja je 1300 bauda, ali vam predlažemo da, kada snimate neki važan program, unesete sledeću naredbu i tako smanjite brzinu na 300 bauda. . .". Bazične memorije računara „galaksija" nije velike, pa snimanje na kasetu ne traje predugo, a verifikacija snimka otklanja sve ostale potencijalne probleme.

Port za proširenja

Na računaru se, najzad, nalazi i standardni port za ekspanziju. Na njega bi mogle da budu priključene „velike" periferijske jedinice, kao što su štampači, ploteri i disk-jedinice, ali ovo verovatno neće biti čest slučaj. Na port, međutim, mogu da se priključe i generatori tona, palice za igru, AD konvertori, kontroleri i druge „sitnice" kojima ćemo u našem časopisu posvetiti posebnu pažnju u toku sledeće godine. Računar „galaksija" će, dakle, biti opremljen periferijskom opremom, koja će biti u skladu sa mogućnostima i željama njegovih

START

DA LI VOLIŠ MIKRORAČUNARE ?
NE: NIJE TREBALO DA ČITAŠ OVO IZDANJE „GALAKSIJE"
DA: DA LI ŽELIŠ DA IMAŠ JEDAN ?
NE: JA TI NE MOGU POMOĆI
DA: VOLIŠ LI DA PRAVIŠ STVARI SVOJIM RUKAMA ?
NE: MOŽEŠ LI DA KUPIŠ RAČUNAR „GALAKSIJA" ?
DA: KUPI RAČUNAR „GALAKSIJA" KOD „ELEKTRONIKE INŽENJERING"
NE: IMAŠ LI MINIMUM ALATA I ISKUSTVA ?
DA: POŠALJI NARUDŽBENICU I NABAVI OSTALI MATERIJAL
DA LI JE STIGAO MATERIJAL ?
NE: U MEĐUVREMENU PROUČI DOBRO ČLANAK O SAMOGRADNJI
DA: SAGRADI RAČUNAR
PROKONTROLIŠI SVE SPOJEVE
IMA LI GREŠAKA ?
DA: ISPRAVI GREŠKE
NE: PRIKLJUČI MONITOR I NAPAJANJE
DA LI RADI ?
DA: IMAŠ ZLATNE RUKE !
POSTAO SI ČLAN PORODICE VLASNIKA RAČUNARA „GALAKSIJA"
PROČITAJ ŠKOLU BASIC-A U OVOM BROJU
SREĆNO !
NE: MOŽEŠ LI SAM DA NAĐEŠ GREŠKU ?
NE: KONSULTUJ NEKOG STRUČNJAKA

Ne očajavajte ako niste vešti u samogradnji

Postoji i rešenja za one koji nisu u mogućnosti da nabave delove i sami sklope ovaj računar Beogradska firma „Elektronika Inženjering" će u saradnji sa Zavodom za učila i nastavna sredstva ponuditi najširem tržištu komercijalni model „galaksije" koji će u svemu biti jednak opisanoj samogradnji.

Kakvog interesa ima „Elektronika Inženjering", koja je uložila ne baš mala sredstva u proizvodnju računara „galaksija", da dozvoli stotinama potencijalnih kupaca da sami sagrade računar umesto da ga kupe u prodavnici? Može da se kaže da je takav stav rezultat poslovne dalekovidosti: kao što je, recimo, poslastičaru stalo da na svetu ima što više ljubitelja slatkiša, tako i proizvođač kompjutera želi da imamo što više ljubitelja računara. Jednog dana će se na našem tržištu pojaviti i daleko složeniji i moćniji modeli računara. Sve će tada biti lakše ako oni dođu na pripremljen teren.
Takođe „Elektronika Inženjering" u takvom potezu vidi i širok društveni interes, kad već nadležni ne mogu ili ne žele da ga vide.

vlasnika. Što više čitalaca ovih redova napravi računar, posvetićemo mu više pažnje i objaviti više šema.

. . . i softverska koncepcija

Na razvoj softvera računara „galaksija" je utrošeno zaista nebrojeno časova rada. Počeli smo od Microsoftovog Level I bejzika, koji je zauzimao četiri kilobajta i pružao mizerne mogućnosti: rada sa alfanumericima nije bilo, programi su mogli da se ispravljaju samo prekucavanjem čitavih linija, nije se moglo programirati na mašinskom jeziku, časovnik nije postojao. Kroz bezbroj izmena od Microsoftovog bejzika su ostali samo potprogrami za sabiranja, oduzimanja, množenja i deljenja brojeva i za realizaciju ciklusa — sve ostalo je promenjeno. Iako ROM i dalje zauzima samo četiri kilobajta, čini nam se da su mogućnosti toliko povećane da je računar „galaksija" postao po mogućnostima uporediv sa nekim računarima koji imaju dve puta veći ROM. Izabrali smo, osim toga, naredbe koje otvaraju veštom programeru mogućnost da na najjednostavniji mogući način dogradi opcije bez kojih smo ga ostavili, jedan od najboljih primera je naredba PTR koju ćete upoznati čitajući uputstvo za upotrebu.

Maksimalno smo podržali rad na mašinskom jeziku i obezbedili dva linka (link je bajt u RAM-u na kome se nalazi instrukcija RTN (vrati se) koju računar s vremena na vreme izvršava ali koju korisnik može da promeni i tako natera kompjuter da „izađe iz ROM-a"). Jedan od njih omogućava jednostavno dodavanje novih naredbi, a drugi daje podršku eventualnom štampaču i omogućava neke druge „specijalne efekte" kojima će se časopis „Galaksija" baviti sledećih meseci.

Na štampanom kolu se, kao što vidimo iz montažne šeme, nalazi još jedan EPROM od 4 kilobajta, kojim u ovom trenutku ne raspolažemo — umesto njega ostavićete jedno prazno mesto. To mesto je namenjeno proširenjima: možda ćemo u budućnosti pripremiti još jedan EPROM u kome će se nalaziti program za rad sa mašinskim jezikom, proširenje postojećeg bejzika ili nešto treće. No, o tome ćemo govoriti kada budemo sigurni da je računar „galaksija" stekao popularnost koju mu mi predviđamo.

Čini nam se da je sadašnja verzija bejzika dobra, iako unekoliko nestandardna. Izbacili smo sve nepotrebne naredbe (čak i tradicionalno LET), dodali neke nove, a neke postojeće preimenovali. Zašto smo uradili ovo poslednje? Jedna od mogućnosti računara je skraćivanje naredbi na jedno slovo i tačku, čime se ne štedi samo RAM nego i vreme izvršavanja programa. Nismo želeli da se pojavljuju naredbe koje počinju istim slovom, pa smo promenili imena nekih od njih. Ipak, strogo smo pazili da se ne pojavi neka naredba koja u bejziku bilo kog računara (koliko samo ima verzija bejzika?) ima neko drukčije dejstvo — ako je bilo potrebno, izmislili smo poneku reč (npr. TAKE).

Pre nego što pređemo na tehnička pitanja izrade računara, jedna izjava zahvalnosti: konstruktor računara „galaksija" se zahvaljuje Dejanu Ristanoviću, čija stručnost i iskustvo su svakako bili presudni u nalaženju zrele koncepcije softvera, kao i Bori Trboviću i Zoranu Vasiljeviću, na drugarskoj pomoći pri realizaciji „hardverskog" dela projekta.

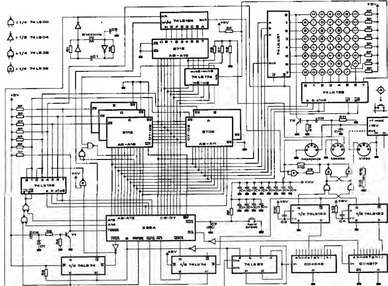
Kako radi „galaksija" prvih nekoliko mikrosekundi

Kao u svakom dobro organizovanom sistemu, i u računaru „galaksija" se zna ko izdaje naređenje, ko je za šta odgovoran i ko čeka da bi „priskočio u pomoć" kad za to dobije zahtev. Varate se ako mislite da je „glavni gazda" mikroprocesor. Iako je on najveći, najkomplikovaniji i troši najviše struje na celoj ploči računara, on ništa ne radi proizvoljno, već samo izvršava naređenja koja mu izdaje ROM (EPROM 2732), tačnije program koji je upisan u njega. Čak ni ROM, međutim, ne izdaje naređenja kad mu se to prohte, već tek kad ga mikroprocesor za to upita. Šta se, zapravo, dešava u računaru „galaksija" od trenutka kad ga uključite u struju?

Recimo da ste tek uključili računar. Sklop tranzistora T1 sa otpornicima R4 i R5, elektrolitskim kondenzatorom C6 i diodom D1 drži nožicu „RESET" mikroprocesora na logički niskom nivou (između 0 i 0,8V) oko pola sekunde, dok se C6 napuni, a zatim naglo tranzistor prestane da provodi i „pull-up" otpornik RP podigne napon iste nožice na logički visok nivo (iznad 2,4V) i tako ostaje sve dok računar radi. To je takozvani postupak inicijalizacije, u kome mikroprocesor dobija naređenje da izbriše sve što se slučajno zateklo u njemu u trenutku uključenja i započne posao ispočetka.

Prva stvar koju će uraditi biće da upita ROM koja je njegova prva naredba. On to praktično čini ovako:

- Svaku od 16 adresnih linija (A0 do A15) postavi na nizak logički nivo (dakle, binarno stanje je 0000 0000 0000 0000, to je adresa 0), nožicu MREQ (Memory Request = zahtev memoriji) spusti na nizak logički nivo (kod mikroprocesora Z80A je zastupljena takozvana negativna logika, znači da je visok logički nivo, koji se predstavlja brojem 1, „pasivno stanje" a nizak, predstavljen nulom, „aktivno stanje"), čime je prozvao memorijsku adresu 0. Iz memorijske mape vidimo da ta adresa pripada ROM-u. Ostaje samo još da saopšti da li to prozivanje predstavlja upis u memoriju ili čitanje iz memorije? Naravno, pošto se ovde traži naredba, radi se o čitanju, dakle nožica RD (read = čitati) će biti aktivna (logički niska). Uostalom, jedna od osnovnih osobina ROM-a je da mikroprocesor može iz njega samo da čita, a ne može da upisuje.
- Dekoder 74LS156, koji ima uvid u „visoke adresne linije" (A11 do A15), a samim tim grubo zna koji je deo iz memorijske mape prozvan, čini aktivnim izlaz 0, koji aktivira nožicu OE (Output Enable=



Krvotok i nervni sistem: Električna shema računara „Galeksija"

dozvola izlaza) ROM-a. Sad ROM zna da je prozvan i uzima preciznu informaciju sa adresnih linija A0 od A11 koji od ukupno 4096 bajta se traži. Nalazi adresu 0 i na osmobitnu magistralu podataka (data bus) stavlja podatak 1111 0011 (heksadekadno F3).

■ Mikroprocesor sa magistrale podataka čita podatak &F3, prosleđuje ga do svog internog dekodera instrukcija i tu doznaje da je dobio naređenje koje se zove DI (disable interrupt = onemogući prekid), što znači: neću dozvoliti da me bilo ko prekine u radu — čak se neću obazirati ni na zahtev da generišem sliku na ekranu. Bar dok ne dobijem suprotno naređenje: EI (enable interrupt = dozvoli prekid).

■ Ubrzo će stići i takav zahtev. Ali, pošto mora sve da se radi po redu (osim ako stigne naredba da se nešto uradi i preko reda — to bi bio mašinski ekvivalent bejzik naredbi GOTO) mikroprocesor uvećava svoj programski brojač i čita naredbu sa adrese 1: tu nalazi podatak 1001 0111 ili &97, što znači SUB A, ili: oduzmi stanja registra A od stanja registre A i rezultat smesti u registar A. Dođavola, kakav je smisao ove suluđe operacije? Da vidimo: bilo kakvo da je stanje registra A, posle ove operacije stanje će biti O. Pa, da li je to onda brisanje sadržaja registra A? Upravo to!

I tako dalje, pedantno i precizno, mikroprocesor će izvršavati svaki zahtev koji dobije od ROM-a (ili čak i od RAM-a, ako koristite naredbu USR), skakaće po memoriji, upisivaće u RAM i čitaće iz njega. Često će tok programa zavisiti i od toga koji ste taster pritisnuli, kakav ste program upisali ili kakav je impuls stigao sa kasete, što na kraju govori da čak ni ROM nije „gazda", već da ste to vi, jer su sve snage računara uperene na to da se što bolje izvrši svaka naredba koja stiže od vas.

Naravno, nije dovoljno da računar radi — potrebno je i da se to vidi na ekranu. Da bi nam bilo jasno kako računar generiše tekst ili grafički oblik na ekranu, potrebno je najpre da razumemo kako uopšte televizor „crta" sliku.

Elektronski top katodne cevi (ekrana) crno-belog televizora stalno emituje tanak mlaz elektrona ka vidljivoj površini ekrana. Ovaj mlaz bi stalno pogađao centar slike, i, osim vrlo svetla tačka na sredini, na ekranu se ništa drugo ne bi videlo kad ne bi postojao takozvani otklonski sistem, koji specijalno prilagođenim promenljivim magnetnim poljem skreće mlaz elektrona da pogađa površinu ekrana sleva nadesno, red po red odozgo nadole, dok ne iscrta celu sliku, a onda sve ispočetka — tako 50 puta u svakoj sekundi.

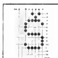
Naravno, pošto na ovako ravnomerno osvetljenom ekranu nema nikakvih informacija, u katodnu cev je postavljena i jedna prepreka, od koje zavisi koliko će elektrona stići do površine ekrana, samim tim i koliko će jak biti svetlosni odziv. Na ovu prepreku se dovodi signal slike (video-signal). Viši pozitivan napon video-signala odgovara svetlijoj površini na ekranu, a niži tamnijoj. Posle svake ispisane linije (što se na računaru „galaksija" događa 16000 puta u sekundi, mada bi po standardu trebalo da bude 15625 puta — ova razlika je zanemarljiva), računar pošalje kratak „horizontalni sink" — signal koji je „crnji od crnog", dakle još niži napon video-signala. Kad se završi ispisivanje poslednje linije na ekranu — računar pošalje znatno duži „vertikalni sink", na isti način. Ove negativne impulse televizor (ili monitor) koristi da sinhronizuje singal otklonskog sistema sa signalom slike. Da bi ispisao ceo red teksta na ekranu, elektronski mlaz mora da povuče 13 horizontalnih linija. Često se za slova koristi samo 9, a preostale 4 linije čine međuprostor između redova.

Širina impulsa horizontalnog i vertikalnog sinka je određena R-C članovima monostabilnih multivibratora koji se nalaze u čipu 74LS123 (za horizontalni impuls „nadležni" su R12 i C3, a za vertikalni R13 i C4). Ovi impulsi su nešto duži nego što propisuje standard, jer je praksa pokazala da se tako dobija stabilnija slika. Ova dva mono-

stabilna multivibratora se pobuđuju iz lanca video-delitelja, to su brojači 74LS93 (učestanost oscilatora deli sa 15), CD4040 (izlaz iz prvog brojača deli sa 1024) i CD 4017 (tako dobijenu učestanost podeli sa 10) Izlazna učestanost će biti 6144000 Hz : 15 : 1024 : 10 = 50 Hz to je tačno učestanost kojom se svaki put iznova generiše slika na ekranu

Jedan od izlaza poslednjeg čipa u lancu video-delitelja (CD4017) se, preko invertora (1/6 čipa 74LS04), dovodi do nožice INT (interrupt = prekid) mikroproossora Z80A. To znači da će svaki put kad je ovaj izlaz aktivan (to se događa uvek kad je elektronski mlaz koji ispisuje sliku negde iznad prvog reda, ako prekid nije internom programskom zabranom DI onemogućen) mikroprocesor izvršiti sledeći postupak

- Registrovaće da je zahtev za prekid posla stigao, ali neće prekinuti dok ne završi tekuću instrukciju
- Kad potpuno završi instrukciju u toku koje je prekinut (to može da traje i do 23 „takta" oscilatora), aktiviraće izlaze M1 i IORQ, čime će „upaliti" flip flop (1/2 čipa 74LS74) i aktivirati sopstveni ulaz WAIT („čekaj")
- Dok je WAIT aktivan, mikroproossor ne radi ništa, jednostavno, miruje i čeka da ponovo postane pasiven (logički visok) To će se dogoditi kod prvog sledećeg horizontalnog sinka izlaz 1/2 čipa 74LS123 obeležen sa $\bar{Q}$ (nadvučeno, što znači inverzni, dakle negativni) izlaz će „ugasiti" preko nožice R flip-flop 74LS74 Ovo je učinjeno da bi dalji rad mikroprocesora, koji će uskoro početi da generiše sliku bio potpuno sinhronizovan sa sinkom
- Mikroprocesor more da zapamti dokle je stao sa poslom Na posebno mesto u memoriji smešta adresu sa koje je, da nije bio prekinut, trebalo de očita sledeću instrukciju kao i stanja svih svojih registara
- Počinje da izvršava program koji postoji u ROM-u počev od adrese &0038 (ne pitajte nas zašto baš od te adrese to znaju samo konstruktori mikroprocesora)
- Upisuje u prva četiri flip-flopa čipa 74LS174 stanje 0001 Ovako memorisano stanje se prosleđuje do karakter-generatora (2716) gde određuje koja linija (od ukupno 13) prvog reda teksta se trenutno ispisuje
- Počinje da proziva stanja memorijskih lokacija &2800 do &281F RAM-a što odgovara prvom redu teksta u video-memoriji
- Sadržaj svake ćelije video-memorije se vodi do karakter-generatora, gde određuje koji će oblik slova biti prozvan Recimo da je to, kao na našem primeru, slovo „Š" (stanje memorijske ćelije je 0101 1110 ili &5E) Već smo rekli da je sada adresirana prva gornja linija na slovima Kod ovog slova upaljeni su bit 4 i bit 8 Pošto je i ovde zastupljena negativna logika izlaz iz karakter-generetora će biti 1010 1111, dakle &AF Ovaj izlaz je „uhvaćen" u pomerački registar 74LS166 Kao na pokretnoj traci, svih 8 bita „upadnu" paralelno a ispadaju jedan po jedan jer ih iz pomeračkog registre isteruje oscilator brzinom od 6144000 bita u sekundi (dakle, crtanje jedne tačke treje 0,00000016 sekundi!)
- Kad se ispiše prva linija, u 74LS174 se upiše binarno stanje 0010 ili &2 (druga



Stotine mikrooperacija za jedno slovo. Kako „Galaksija" crta slovo „Š"

```
SPECIFIKACIJA DELOVA
ZA RAČUNAR „GALAKSIJA"

OTPORNICI (1/8 W)

R1     910 OMA
R2     910 OMA
R3     330 OMA
R4     10 K
R5     10 K
R6     8.2 K
R7     8.2 K
R8     1 K
R9     62 OMA
R10    50 OMA
R11    2.4 K
R12    390 OMA
R13    27 K
R14    100 OMA
R15    VIDI PRIMEDBU
R16    VIDI PRIMEDBU
R17    220 OMA
RP     1.5 K do 10 K (17 KOMADA)

PRIMEDBA: VREDNOSTI OTPORNIKA R15 I
   R16 ZAVISE OD TIPA UPOTREBLJENOG
   RF MODULATORA. NAJČEŠĆE ĆE BITI
   DOVOLJNO DA SE R15 IZOSTAVI,A
   R16 ZAMENI KRATKOSPOJNIKOM. AKO
   JE SIGNAL IZOBLIČEN, POKUŠATI
   SA R15=R16=62 OMA

KONDENZATORI

C1     5 nF
C2     100 nF
C3     5 nF
C4     100 nF
C5     10-30 pF (VIDI TEKST)
C6     10 µF ELKO
C7     1 µF ELKO
CN     47 DO 200 nF (13 KOMADA)

KVARC - KRISTAL

6144  KHz

TRANZISTORI

T1 i T2   BC 107-109 (2 KOMADA)

DIODE

D1     1N4148 ili slicna
LED    svetleća dioda

INTEGRISANA KOLA

74 LS 00
74 LS 04
74 LS 32   (VIDI TEKST)
74 LS 38
74 LS 74
74 LS 93
74 LS 123
74 LS 156   (2 KOMADA)
74 LS 166
74 LS 174
74 LS 251
CD 4017
CD 4040
Z80A CPU
2716 EPROM
2732 EPROM
6116 C-MOS RAM (1 do 3 komada)

TASTATURA

55 TASTERA TX ILI TY - 54 KAPICE
             PREMA MONTAŽNOJ SHEMI
MASKA ZA TASTATURU (samonoseca)

TASTER

POVRATNI - ZA  HARD-BREAK

DŽEKOVI

DIN - PETOPOLNI (3 para)
```

linija), što će iz karakter-generatora prozivati drugu liniju svih slova Kad naše slovo Š stigne na red, izlaz iz karakter-generatora će biti 1101 1111 ili &DF

- Tačku po tačku, liniju po liniju, red po red — iscrtava se slika na ekranu Naše oko je presporo da bi videlo ovu postupnost i sve se zato sliva u jednu stabilnu sliku
- Kad se završi ispisivanje poslednjeg reda na ekranu, mikroproossor iz memorije vraća svojim registrima prethodne stanja, i na kraju potprograma za generisanje slike nalazi instrukciju RETN — poslednji znak kojim daje na znanje da programskom brojaču dodeljuje adresu na kojoj je bio u trenutku kad je prekinut u redovnom poslu i sve nastavlja kao da se ništa nije dogodilo Taj njegov „mir" će potrajati samo dok elektronski mlaz ispisuje donji, a zatim i gornji prazan deo ekrane, kad se opet preko nožice INT, stići novi zahtev da se prekine posao i da se počne sa ispisivanjem nove slike

Sledeći važan podsklop računara je tastatura Matrični spoj (povezivanje u vrste i kolone) je vrlo ekonomičen, jer zahteva minimum čipova za sprezanje sa mikroprocesorom Dekoder 74LS156 (pazite, u računaru „galaksija" postoje dva takva čipa) proziva vrste, a ako pritisnemo neki od tastera, nizak logički nivo (negativna logika!) prozvane vrste se prenosi na određenu kolonu, gde promenu očita selektor 74LS251 i prosledi je do linije DO mikroprocesora Ako u toku skaniranja tastature mikroprocesor naiđe na ovakav slučaj (neki od ulaza selektora je postao nizak) on će tu informaciju proveriti još 256 puta, de bi bio potpuno siguran da se ne radi o lažnom impulsu, a onda će iz izraza VRSTA × 8 + KOLONA sračunati koji je taster pritisnut

Interesantno je da, dok vi upisujete neki program, mikroprocesor bar 99% ukupnog vremena provodi skanirajući tastaturu (ne računajući, naravno, vreme generisanja video-signala), jer kad vi pritisnete neki taster, on ga samo upiše u video-memoriju i u bafer i odmah se vraće na dalje skaniranje tastature, ne razmišljajući o tome šta ste napisali

Izuzetak je jedino eko ste pritisnuli neki od specijalnih tastera — BRK, (shift) DEL ili LIST, jer onda ignoriše sve što je doted upisano u bafer i izvršava naredbu odmah Ali, evo kakvu ste lavinu pokrenuli kad pritisnete taster ENTER

- Računar se vraća na početak bafera i, ignorišući blankove (prazna mesta), testira da li je prvi upisani znak cifra ili slovo
- Ako je cifra — tu onda mora biti broj programske linije, dakle, konvertovaće decimalni broj u binarni oblik i pretražiće u memoriji da vidi da li već ima programsku liniju pod istim brojem Ako ima, izbrisaće je (pomeriće sav program iza te linije unazad, za onoliko mesta koliko iznosi dužina te linije) Nakon toga će sračunati dužinu teksta u baferu, napraviti toliko mesta u memorisanom programu (pomeriće sve linije sa većim brojem nagore u memoriji) i premestiti bajt po bajt, upisanu liniju iz bafera na oslobođeno mesto Još uvek ga ne zanima šta ste upisali — zasad samo memoriše
- Ako je slovo — shvatiće da je u komandnom načinu rada i da od njega očekujete de odmah izvrši instrukciju (ili grupu instrukcija) Ta instrukcija može biti, recimo PRINT 5+5, ali i RUN — u tom slučeju ste inicirali izvršenje celog memorisanog programa Reč po reč trudiće se de prepozna vaše naredbe, a onda će odlaziti na potprograme u ROM-u koji mu objašnjavaju kako da izvrši svaku od njih

Činjenica da ste bili dovoljno motivisani da pročitate ovaj tekst, koji se bavi ne baš jednostevnim pitanjima, govori o tome da spadate u grupu ljudi koji su u stanju da se uhvate ukoštac sa logičkim problemima Potrudite se da dođete do literature koja obrađuje ovu problematiku — u poslednje vreme ima i kod nas odličnih knjiga na tu temu I časopis „Galaksija" će vam stalnom rubrikom pomoći u tome

Samogradnja računara „galaksija" u stripu

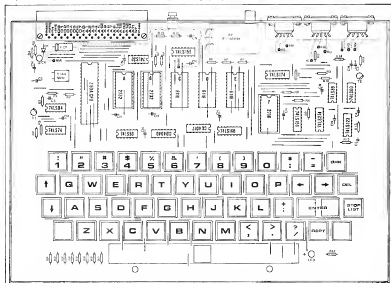
Evo nas, konačno, i na praktičnom delu posla. Očekuje nas ozbiljan ali prijatan rad, koji će biti nagrađen nesvakidašnjim zadovoljstvom što smo stvorili i oživeli jedan ovako inteligentan uređaj. Nemojte se obeshrabriti ako smatrate da nemate dovoljno iskustva: to je prvi i dobar znak da imate samokritičkog duha, a on vam je, verujte, u ovom poslu potrebniji od iskustva. Zastanite posle svakog, i najmanjeg i naoko beznačajnog detalja, i procenite da li je to dobro urađeno i — „galaksija" će proraditi iz prve!

Važne odluke

Pre početka rada treba doneti nekoliko važnih odluka. Prvo, da li želimo da ovakav sistem bude konačan ili ćemo ostaviti mogućnost da ga u budućnosti proširujemo dodavanjem štampača, više memorije, programatora, „muzičke kutijice", i slično. Ako ne želimo ova proširenja — uštedeli smo višepolni konektor i jedno integrisano kolo (74LS32, koje ćemo zameniti jednim kratkospojnikom obeleženim crticama na montažnoj shemi). Ako ste u nedoumici — mi vam savetujemo da ipak ugradite ova dva dela, mada za to ni posle neće biti kasno.

Drugo pitanje je da li ćemo se opredeliti za nemodulisan video-signal ili modulisan (RF) signal slike. Nemodulisan video-signal ne zahteva ugradnju RF modulatora u računar i daje stabilniju i kvalitetniju sliku, ali se zato ne može priključiti na bilo koji televizor — neophodno je imati specijalni monitor ili crno-beli televizor sa dograđenim monitorskim ulazom. Ovo ne zahteva nikakva dodatna ulaganja, ali je neophodno imati predznanja i iskustvo u radu sa TV prijemnicima. Dalje, takav televizor mora biti tranzistorski (cevni ne dolaze u obzir) i mora imati mrežni transformator (a ne takozvanu „vruću šasiju"), najčešće su oba ova uslova ispunjena kod malih prenosnih crno-belih televizora kod kojih postoji spoljni priključak na akumulator od 12 V. Neke savete za dogradnju monitorskog ulaza na ovakav televizor ćemo opisati u daljem tekstu. Ali, ako ugradimo RF modulator, bićemo oslobođeni svih ovih problema i moći ćemo da se priključimo na antenski ulaz bilo kog televizora.

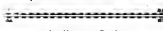
Moraćemo, takođe, da odlučimo koje čipove ćemo smestiti na podnožje, a koje lemiti direktno na štampano kolo. Savetujemo vam jedino da za EPROM-e (2716 i 2732) koristite podnožja, a za ostalo se opredelite sami. Prednost podnožje je u tome što smanjuju rizik de upropastite neki čip i što je zamenom vrlo lako lokalizovati neispravan integralac (naravno, ako takvog uopšte ima, odnosno ako eventualne krivica nije do neke druge komponente), jer je razlemljivanje čipova izuzetno osetljiv posao. Podnožja, na žalost, ako nisu vrhunskog kvaliteta, lošim kontaktima češće prave probleme nego bilo koje druge komponente. Da bi bilo pouzdano, podnožje mora da bude vrlo kvalitetno, a to ponekad znači da je skuplje i od samog čipa.



Montažna shema. Raspored elemenata u računaru „Galaksija"

GALAKSIJA

Štampano kolo u razmeri 1:2: Zbog visokog profesionalnog kvaliteta i pristupačne cene komercijalne pločice njena samogradnja se ne isplati



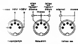
RASPORED PRIKLJUČAKA NA KONEKTORU

1	N.C.	12	MASA	23	D 0	34	A 3
2	N.C.	13	MASA	24	D 1	35	A 4
3	N.C.	14	MASA	25	D 2	36	A 5
4	N.C.	15	MASA	26	D 3	37	A 10
5	MASA	16	WR-	27	D 4	38	A 9
6	MASA	17	A 15	28	D 5	39	A 8
7	MASA	18	A 14	29	D 6	40	A 7
8	MASA	19	IORQ-	30	D 7	41	A 6
9	MASA	20	M1-	31	A 0	42	A 12
10	MASA	21	MREQ-	32	A 1	43	A 13
11	MASA	22	MASA	33	A 2	44	A 11



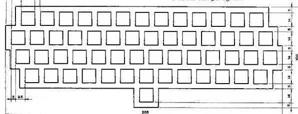


Dvostruka štampa: Konektor za proširenja u obliku štampanog kola



Veza sa spoljašnim svetom: Priključci i raspored izvoda na zadnjoj strani „Galaksije"

***Maska za tastere:** Definitivan oblik zavisi od tipa mehanizma za razmaknicu i zato pre izrade treba sačekati isporuku tastature, oni koji naruče tastaturu u prvom krugu ne moraju ni o čemu da brinu — delovi u kompletu će savršeno odgovarati jedni drugima*



Srce računara „Galaksija" Mikroprocesor Z80A i EPROM 2732 sa bejzik interpraterom

1 Pred nama je materijal koji smo sakupili sa toliko muke i iz koga će za nakoliko časova da izraste računar galaksija'' U dnu slike lako prepoznajemo tastera i kapice tastera sa utisnutim oznakama desno su otpornici (svi su snage 1/8 W mada mogu da se koriste i otpornici veće snage) levo kondenzatori a u sredini čipovi (integrisane kola) Posebnu pažnju treba obratiti na MOS i CMOS čipove.

2 Pošto je štampano kolo jednoslojno biće nam potrebno doste kratkospojnika. Njih je najlakše izraditi od puna bakarne žice izvađene iz popularne plavo-bele telefonske parice Olakšavajuća okolnost je što su dužine standardizovana na 5 10 20 30 i 40 mm pa je lako izrezati alatku za njihovo precizno savijanje (pri izradi ove jednostavne alatke treba voditi računa o prečniku žice

3 Sklapanje računara započinjemo postavljanjem prvog kratkospojnika, pažljivo gledajući montažnu shemu Neki kratkospojnici prolaze ispod čipova ovo neće praviti probleme ako su kratkospojnici pedantno savijeni i ako leže uz samo štampano kolo Pažnja! Ovo je pogled sa strane elemenata a na kako se može učiniti sa strane vodova!

4 Kada okrenamo ploču da bismo zalemili prvi kratkospojnik postaje nam jasno zašto montaža počinja od najnižih komponenata Da smo na primer počeli od tastera sve niže komponente bi prilikom docnijih lemljenja ispadale Ako nikada niste lemili dobro je da najpre malo eksperimentišete na nekoj drugoj pločici Vrh lemilice treba da buda dobro oblikovan turpijom očišćen i kalajisan Lemi se tako što se sa jedne strane prinese tinol žice, a sa druge dobro zagrejani vrh lemilice Treba paziti da tinola na lemnom mestu na satena previše Ma koliko to paradoksalno zvučalo u protivnom ćemo dobiti loš električni kontakt



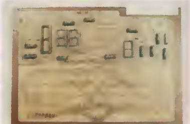
5 Svi kratkospojnici su postavljeni i zalemljeni. Pažljivo ih prebrojmo, treba da ih bude tačno 119. Ukoliko na vašem štampanom kolu neki nedostaje, moraćete ponovo da konsultujete montažnu shemu. Obratimo pažnju na čip 74LS32: kao što smo rekli u uvodu, možemo ga zameniti kratkospojnikom (isprekidana linija na montažnoj shemi) ako ne želimo proširenja sistema preko konektora. To će onda biti 120-ti kratkospojnik.



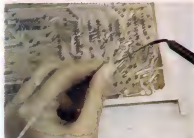
6. Sledeća faza je montaža otpornika, koja je u mnogo čemu slična montaži kratkospojnika dužine 10 mm.



7 Kod montaža čipova, koja je sledeća na redu, izuzetnu pažnju treba obratiti na orijentaciju, jer se i iskusnim profesionalcima dešava da okrenu čip naopako. Neki čipovi su obeleženi polukružnim usekom kao na montažnoj shemi a drugi ugraviranom tačkom pored nožice broj 1. Napominjemo da natpis na čipu nije baš uvek okrenut tako da počinje od prve nožice. Pošto će na „galaksijinom" štampanom kolu sa gornje strane biti odštampan raspored elemenata, ovde ne bi trebalo da bude nikakvih problema.



8. Čipovi su postavljeni, ali ne svi — zasad su izostavljeni već pomenuti MOS i CMOS tipovi CD 4017, CD 4040, 6116, 2716, 2732 i Z80A. Najbolje je da ih ostavimo za kraj, ali nema razloga da ne stavimo podnožje. Sada je trenutak da pre lemljenja još jednom proverimo da li je svaki čip na svom mestu i pravilno okrenut. Nije slučajno što ovaj savet ponavljamo, svako nestrpljenje i neopreznost prilikom montaže skupo se plaćaju u trenutku prvog uključenja.



9. Lemljenje čipova je posebno osetljiv posao, jer su međusobna rastojanja nožica svega 2,54 mm, a često između njih prolazi i vod. Ako se dogodi da se nepažnjom napravi neželjeni most od tinola, skinućemo ga tako što ćemo na istom mestu rastopiti još (svežeg!) tinola, pa onda sve odstraniti u jednoj kapljici vrhom lemilica.



10 Kondenzatori su sledeći po visini. Montirajmo dakle i njih. Najbolje je koristiti takozvane disk-kondenzatore, jer su najmanjih dimenzija i najjeftiniji, ali ako ima problema kod nabavke — koristite onakve kakve imate. Kapacitet svih kondenzatora obeleženih slovom C nije kritičan, a još manje njihov probojni napon. Kondenzator C5 nećemo još montirati. Najverovatnije neće biti ni potreban, ako imamo odgovarajući kvarc. Kad stignemo do puštanja u pogon, biće više reči o tome.



11 Tu su i dve tranzistora NPN tipa male snage uz levu i desnu ivicu ploče po jedan Malo pažnje i kod montaže nećemo pogrešiti ako pogledamo tranzistor odozdo videćemo da su mu nožice razmeštene kao da su na uglovima pravouglog ravnostranog trougla Isto su razmeštena i rupice za tranzistor na štampi U levom gornjem uglu štampana ploče je i jedna mala dioda Najčešće je katoda (koja je bliža sredini štampanog kola) obeležena jednim prstenom po obimu cilindričnog kućišta



12 Uzbuđenje svakako raste stigli smo do tastature Bez obzira da li smo masku sami izrezali od vitroplasta ili aluminijumskog lima (što ne bismo preporučili čak ni najljućem neprijatelju) prema našem crtežu, ili smo je naručili i dobili zajedno sa tasterima, ona nam je neophodna bez nje bi se svaki taster klatio za sebe i verovatno bi se kapice češale jedna o drugu Maska je samonoseća — nigde se, dakle, ne pričvršćuje za štampano kolo.



13. Prvo ćemo u ivične otvore maske staviti nekoliko tastera zasad bez kapice a onda ih zalemiti tako da maska stabilno stoji Obratimo pažnju da tasteri ne stoje naopako na montažnoj shemi se vidi da su izvodi okrenuti ka nama Kratkospojnici neće smetati jer su postavljeni tačno između tastera Dalje će ići lako postoji ukupno 56 tastera i svi su jednaki



14 Pošto je red sa lemilicom priveden kraju zalemićemo ili postaviti u podnožja MOS i CMOS čipove Pažnja — ovi čipovi su veoma osetljivi na statički elektricitet Svakako je dobro prvo proučiti članak „opasna krivina



15. Klik — klik — klik! Kapice tastera su na svojim mestima, i sad već čitava stvar poprima ozbiljan oblik Skoro da nas mami pa da počnemo da pišemo program Ali strpljenja strpljenja



16 Zapazićemo da je jedna kapica tastera (sa oznakom RET i ENTER što je isto), dvaput šira od ostalih Ona se montira na dva tastera Ako pažljivo pogledamo stazice na štampanom kolu videćemo da su kontakti te dva tastera spojeni paralelno Funkciju dakle ima samo jedan taster a drugi je tu samo iz mehaničkih razloga.

17 Izbor utikača („džekova") ćemo prepustiti vama Možete upotrebiti onakve kakve imate, ako su bar tropolni Nama se čini da su standardni petopolni DIN-utikači sasvim upotrebljivi lako se nabavljaju (proizvodi ih EI) nisu skupi a za divno čudo — vrlo su pouzdani Obzirom da imaju po pet kontakata predlažemo raspored priključaka dat na montažnoj shemi Dobra osobina ovakvog rasporeda je što slučajnom zamenom džekova nećemo napraviti havariju

18 Pošto kod nas nije baš lako pronaći višepolni konektor štampu smo prilagodili tako da je moguće montirati više različitih tipova konektora ako imaju standardni korak od 2.54 mm Kao najpovoljnije rešenje mi smo odabrali dodavanje još jedne male dvoslojne štampane pločice koja je tako projektovana da na nju može da se priključi višežilni kabl sa 44-polnim „EDGE" („ivičnim") konektorom jer je takav tip najlakše nabaviti, a i cena mu je pristupačna.

19 Naravno sad ćemo, kao što se radi i u proizvodnji napraviti finalnu kontrolu celog štampanog kola prosvetlićemo ploču jakim svetlom izbliza i sa lemne strane vrlo pažljivo posmatrati svaku liniju Minijaturni „mostići" od tinola su česta pojava. Pogledajte zaokružen deo slike — mi smo na našoj štampi našli ne baš tako sitan most od tinola, koji je ko zna kako nastao na tako širokom prostoru između dve staze

20 Naš trud je nagrađen ovim lepim prizorom — čistim i urednim štampanim kolom u uređaju koji će umeti da nam višestruko uzvrati za uložen napor i strpljenje „Galaksija" će raditi za vas bolje od mnogih drugih elektronskih uređaja u ovom veku elektronike ispoljavajući osobinu koju ćemo po prvi put sresti kod jedne naprave — ona će komunicirati sa nama na takav način da ćemo imati utisak da je postala član porodice Zaista nije neobično što mnogi svoj računar smatraju svojim prijateljem

Pročitajte i ovo

opasne krivine

Ako za sobom imate dosta sagrađenih uređaja (koji su uz to još i proradili), svakako se nećete baš doslovno pridržavati svih naših uputstava Ipak, postoje pravila koja ne smete prekršiti jer biste time sigurno izazvali trajna oštećenja komponenata. Nabrojaćemo najbitnije

— Kratak spoj između pozitivnog i negativnog voda za napajanje računara će oštetiti stabilizator 7805. Neki proizvođači ugrađuju automatsko strujno ograničenje u ovaj čip, ali to nemojte da proveravate isto tako, slučajna zamena pozitivnog i negativnog voda od ispravljača do računara će sasvim sigurno biti fatalna za sve čipove.

— Skoro svi čipovi u računaru „galaksija" imaju radni napon od +5V, pri čemu su dozvoljena odstupanja od ±0,25 V integrisana kola će preživeti šokove do 7V, dok su prekoračenja ovog napona opasna.

— Kratak spoj bilo kog izlaza TTL kola (to su čipovi serije 74LS ...) sa pozitivnim vodom za napajanje će trajno oštetiti to kolo. Kratak spoj izlaza sa masom je bezopasan, i možemo ga slobodno primenjivati prilikom eksperimentisanja Ovde treba samo paziti da se ne dogodi da veći broj izlaza istog čipa bude spojen sa masom istovremeno

— U slučaju loše sinhronizacije slike na ekranu monitora, eksperimentisaćemo sa različitim vrednostima otpornika R12, R13, R9 i R10 Nema nikakvih opasnosti ako R12 ili R13 nisu manji od 330 oma, i ako R10 nije manji od 40 oma

— Priključivanje monitorskog izlaza (bez RF modulatora) na TV prijemnik sa „vrućom šasijom" je opasno ne samo za čipove, već i za vaš život. Zbog velike važnosti, ovoj temi smo posvetili poseban tekst „Jednostavan zahvat, fantastični efekti".

— Pošto su MOS i CMOS čipovi vrlo osetljivi na statički elektricitet, potrebno je pažljivo rukovati s njima. Verujući da je većina konstruktora već upoznata sa tehnikom rada sa ovim čipovima (u računaru „galaksija" to su CD4017, CD4040, 2716, 2732, 6116 i Z80A), navešćemo samo nekoliko osnovnih saveta

— Poželjno je koristiti uzemljenu lemilicu Ako nemamo takvu, možemo se poslužiti običnom, ako hladniji kraj metalnog dela lemilice (bliže ruci) obavijemo nekoliko puta bakarnom žicom, čiji drugi kraj spojimo sa uzemljenjem na šuko-utičnici.

— Ako u prostoriji u kojoj radimo imamo sintetički tepih, statički potencijal našeg tela u odnosu na zemlju može da dostigne čak 300 volti! To nas ne ugrožava mnogo, jer će se taj naboj „isprazniti" za vrlo kratko vreme kad dodirnemo neki uzemljeni predmet ali ako se isprazni kroz nožicu MOS ili CMOS čipa — verovatno će ga učiniti neupotrebljivim Zato se takvi čipovi čuvaju u takozvanim anti-statičkim cevima, a mogu biti i uteknuti nožicama u specijalni provodni sunđer ili jednostavno umotani u staniol

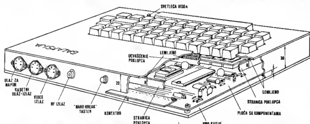
— Naši čipovi će biti potpuno sigurni u toku lemljenja ako napravimo još nekoliko namotaja neizolovane žice oko dela lemilice koji držimo rukom, a drugi kraj žice spojimo sa uzemljenim metalnim delom Tako smo i mi, pošto dolazimo u dodir sa čipom, na istom potencijalu.

— Kad jednom ugradimo čip, on više nije toliko ugrožen, tako da se po završetku montaže možemo osloboditi svih mera predostrožnosti

Izrada kutije računara

konac delo krasi

Mehaničku koncepciju kutije prepuštamo vama, ali ćemo vam dati i jednu ideju pošto na obodu osnovnog štempanog kola ima dovoljno bakra, stranice sa mogu iseći od istog takvog vitroplasta i jednostavno zalemiti ze ploču sa komponentema. Tako štempene ploča posteje mehenički osnov cele kutije, za šta vitroplast zadovoljava i najstrožije mehaničke zehteve





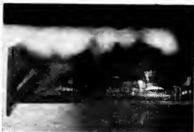
1 Pažljivo ćemo isplanirati dimenzije svakog dela kutije na papiru. Moramo tačno znati koja stranica preko koje prelazi na sastavima. Delovi se lako i precizno isecaju popularnim OLFA skalpelom zasecanjem linije sa obe strana ploče. Posle toga, ako su žljebovi dovoljno duboki, lako je slomiti ploču po zasečenoj liniji. Posle ovakvog sečenja finom turpijom treba obraditi ivice. Ivice koje se lema obrađuju se ravno, a slobodna ivice zaobljeno



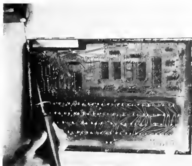
2 Najpre treba obeležiti i očistiti tvrdom gumicom ili finim brusnim papirom sve spojne površine koje ćemo lemiti. Zatim ćemo dobro zagrejati lemilicu od 24 ili 30 W i kalajisati očišćene površine. Biće lakše ako koristimo i pastu za lemljenje



3 Pre lemljenja celog sastava zalemićemo stranicu samo u nekoliko tačaka. Tako ćemo moći pažljivo da izvršimo kontrolu i eventualne korekcije. Treba znati da je jednom zalemljenu stranicu kutija praktično nemoguće razlemiti bez oštećenja



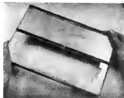
4 Kod lemljenja stranica treba obratiti pažnju na skupljanje legure kalaj-olovo pri hlađenju: ako želimo prav ugao, postavićemo ploče pod tupim uglom (gledano sa strane sa koje se lemi, na slici je to donja strana) jer će posle lemljenja tinol „povući" ploče jednu prema drugoj. Tako ćemo posle hlađenja dobiti prav ugao





6 Na unutrašnju površinu poklopca ćemo zalemiti nekoliko stranica visine oko 10 mm koje mogu da se podese da tesno ulaze u stranice kutije Zato posebno učvršćenja poklopca za kutiju nije ni potrebno

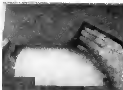
5 Posle stroge provere međusobnog položaja i ugla zalemićemo ceo sastav dve površine Verovatno će biti potrebno da posle svakih nekoliko centimetara sačekamo da se rashlađeni vrh lemilice ponovo zagreje Možda bi ovej problem bio rešen malo jačom lemilicom ali je to pomalo opasno rešenje pregrejani bakar se odlepljuje od vitroplasta



7 Da bi poklopac bio otporniji ne savijanje zalemićemo jednu traku od vitroplasta i kroz sredinu Ostalo nam je još samo dno kutije — možemo ga napraviti od bilo kog materijala koji ne provodi struju Mi ćemo dati prednost ploči od pleksiglase debljine oko 4 mm koju ćemo pričvrstiti za glavnu ploču sa četiri zavrtnja M3 sa kontra navrtkama ili distancerima za spajanje dve površine na rastojanju



8 Ako želite da obojite kutiju i ispišete sve potrebna oznaka — i tu vam možemo pomoći dobrim savetom Postoji naime postupak koji ima sve dobre osobine sito-štampe daje estetski dobre rezultate Ima veliku mehaničku otpornost a može se lako izvesti u amaterskim uslovima Treba da pripremimo dva auto-lak spreja (najbolje da jedan bude beli a drugi tamniji recimo medio-plavi broj 469) bočicu benzine za čišćenja i lithoset-slova i eventualno linije



9 Neophodno je da finim brusnim papirom obrusimo celu površinu koju ćemo obojiti Nigda ne sme da buda sjajna jer bi sa takvih mesta boja brzo otpala Dobro ćemo je očistiti i odmastiti benzinom



10 Ravnomarno ćemo naprskati površinu svetlijom bojom (najbolje belom) Biće korisno ako proučimo uputstvo sa bočice spreja Ovaj sloj treba da se suši najmanje tri časa ali ne na hladnom ili vlažnom vazduhu



11 Lithoset-slovima ćemo preko tek osušene površine ispisati sve potrebne tekstove Ako izvučemo i linije po obodu kutije i pored otvora za tastaturu dobićemo lepši izgled Čistim i suvim prstom ćemo pritisnuti svako slovo da bismo bili sigurni da je dobro zalepljeno

13 Posle oko jednog časa sušenja (ne mnogo duže!) pažljivo ćemo noktom izgrebati slova i linije Možda će posle ova faze rada poklopac izgledali pomalo neprecizno i neuredno Ne obraćajmo zasad pažnju na to



12 Pažljivo ćemo sve to preprskati tamnijom bojom Ovaj sloj trebe da buda što ravnomerniji i tanji tek toliko da se ne providi bela boja



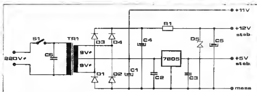
14 Kad na čistu krpicu ili papirnu maramicu stavimo malo benzine za čišćenja i protrljamo površinu bićemo iznenađeni veoma lepim izgledom slova i linija

To je postupak za dobijanja belih slova na tamnijoj površini Možda ste se zapitali da li je moguće upotrebiti najpre tamniju pa onda svetlju boju da bi se dobila tamna slova na svetloj podlozi? Moguće je ali uz opasnost da rezultat ne buda najbolji Gornji sloj mora tada da buda debeo pa će ivice slova biti iskrzane

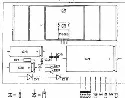


Bez ovog se ne može

ispravljač i stabilizator za napajanje



Električna shema ispravljača



Montažna shema ispravljača

SPECIFIKACIJA DELOVA ZA ISPRAVLJAČ

OTPORNIK

R1 1 K

KONDENZATORI

C1	3300-6800 µF	min.16 V
C2	0.2 do 1 µF	
C3	0.2 do 1 µF	
C4	500 µF	min. 30 V
C5	100 µF	min. 16 V
C6	100-200 nF	min. 400 V

DIODE

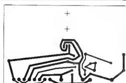
D1	1N5400
D2	1N5400
D3	1N4001
D4	1N4001
D5	canar dioda BZ 12

INTEGRISANO KOLO

stabilizator 7805

TRANSFORMATOR

2 X 9 V min 6 W



Štampano kolo ispravljača

Odmah ćemo reći da se stabilisani napon 12 V koristi samo za napajanje RF modulatora i da ga možete izostaviti ako ne ugrađujete modulator ili imate takav koji se napaja naponom 5 V. Time biste uštedeli komponente D3, D4, D5, C4, C5 i R1.

Kondanzator C6 na primarnoj strani mrežnog transformatora služi za eliminisanje neželjenih smetnji koje bi se mogle pojaviti iz mreže. Ispravljač je punotalasni, i na elektrolitskom kondenzatoru C1 se dobija oko 11 V ispravljenog i filtriranog napona. Integrisani stabilizator 7805 obezbeđuje oko 1 A struje pri naponu od 5 V. Dobro je upotrebiti i transformator koji može da napaja strujom te jačine, bez obzira što računar troši svega oko 0,4 A. Ostatek struje neka služi za kasnije napajanje eventualnih proširenja. Kondenzatori C2 i C3 osiguravaju 7805 protiv oscilovanja.

Pošto stabilizator 7805 u toku rada oslobađa veliku količinu toplote, potrebno ga je montirati na hladnjak. Ako nemamo fabrički, možemo ga napraviti od tri komada aluminijumskog lima dimenzija 35×80, 35×110 i 35×140 od kojih se svaki na dve mesta oštro savije u obliku slova U. Otvor na metalnoj zastavici stabilizatora je za zavrtanj M3, kojim se on dobro stegne za hladnjak. Preporučljivo je pre montaže dodirnu površinu stabilizatora namazati sa malo silikonske paste radi boljeg odvođenja toplote. Nikakvi liskunski izolatori nisu potrebni.

Izaberite sami u kakvu kutiju ćete montirati ovaj ispravljač i transformator. Poželjno je da ima otvora za hlađenje, i ako je metalna, obavezno treba mrezni napon dovesti trožilnim kablom sa „šuko-utikačem". Žuto-zelani vod kabla se sa jedne strane spaja sa lističem za uzemljenja šuko-utikača, a sa druge za masu metalne kutije i minus-pol ispravljača.

jednostavan zahvat - fantastični efekti

Da bismo običan crno-beli televizor pretvorili u monitor, moramo da poštujemo jedno važno ograničenje: video ulaz može da se doda samo televizoru koji ima mrežni transformator. TV prijemnici sa „vrućom šasijom" su vrlo opasni za prepravke jer su galvanski spojeni sa računarom i tako ugrožavaju život onoga ko upravo radi sa njim.

Kako da proverite da li vaš televizor ima „vruću šasiju"? Ako nemate dovoljno iskustva i predznanja, odustanite od tog posla ili ga prepustite stručnjaku. Ako ste sigurni u svoje znanje, otvorite televizor i uključite ga u mrežu (to je ono što, prema uputstvima proizvođača, „nikade ne smete da radite") nikako ne dodirujući njegove metalne delove. Izmerite potencijal mase televizora u odnosu na zemlju. Isključite mrežni utikač, okrenite ga za 180 stepeni pa ponovite merenja. Ako ste u bilo kom slučaju očitali neki napon, zatvorite televizor i odustanite od dalje prepravke. Rešenje vašeg problema se zove RF modulator.

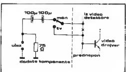
Ako ni u jednom slučaju niste registrovali napon, možete da nastavite sa proverom. Otpor između bilo kog pola mrežnog priključka televizora i mase mora da bude beskonačno veliki (meri se, naravno, sa isključenim napajanjem). Ako je i ova provera dala pozitivan rezultat, imate „zeleno svetlo" za prepravku.

Najpre nabavite shemu vašeg TV prijemnika, jer bez nje nema smisla. Pronađite ulaz u prvi stepen video-pojačavača. Tu je obeležen napon takozvanog „belog nivoa", a sink je 2 volta ispod toga. Tranzistorski TV prijemnici najčešće imaju „beli nivo" na +3 V, a sink na +1V.

Ostavljajući prednapon iz razdelnika priključen na bazu tranzistora, otkačite vod koji dovodi signal iz video-detektora i povežite ga prema našoj slici. Potrebno je da dodate jedan bipolarni elektronski kondenzator od oko 50 µF ili, pošto se bipolarni elektroliti teško nabavljaju, dva elektrolita od po 100 µF koje vezujete kontra-redno (plus polovi jedan prema drugom, a minus polovi su za utičnicu) i prekidač koji služi za izbor funkcije televizora, ne odričemo se, dakle, ni TV prijemnika). Na zadnjoj ploči televizora izbušite otvor za montažu prekidača i utičnice za video-signal.

Za povezivanja je dobro koristiti što kraće vodove koji, po mogućstvu, treba da budu oklopljeni („širmovani") ili bar da im parice budu spiralno uvijene, jedan kabl oko drugog. Ista preporuka se odnosi i na kabl koji povezuje računar i novi monitor.

Time je prepravka završena. Zatvorite televizor i spojite ga sa računarom. Kada ih uključite, biće verovatno potrebno određeno podešavanja horizontalne i vertikalne sinhronizacije, kao i podešavanja televizora na najjači kontrast, pri kome se slova još ne „razmazuju".



Prvo uključivanje *bez panike, sve će biti u redu*

Najpre uključite u mrežu samo ispravljač izmerite napone stebilisani napon od 5 V ne sme da odstupa više od ±0,25 V. Za 12 V (napon koji je potreban za neka RF modulatore) odstupanja mogu da budu i ±1 V. Pošto ste se uverili da su naponi u dozvoljenim granicama, spojite mase ispravljača i računara komadom žice, merni instrument podesite na najširi opseg merenje jačine struje, pa plus pipkom instrumenta dodirnite +5V izlaz ispravljača, a minus pipkom ulaz za +5V na računaru instrument treba da pokaže struju između 300 i 500 mA. Ako je dobijena vrednost u ovim granicama, uklonite instrument sa +5 V i ponovite isto sa +12 V. Zavisno od tipa upotrebljenog RF modulatora (on se jedini napaja strujom koju merimo), otklon kazaljke treba da buda nekoliko miliampera. Da bismo ga registrovali, dakle, moremo da smanjimo opseg instrumenta

Ako je sve u redu, skionimo merni instrument i priključimo monitor preko video-ulaza (ili TV prijemnik preko antenskog), povežimo ispravljač sa računarom i uključimo ga. Ako koristimo RF signal i TV prijemnik, praći ćemo skalu televizora na sve tri opsege da bismo našli gde je prijem najbolji. Računar će napisati prvu reč u svom životu — „READY" (spreman)

Važno je da proradi — ne mora iz prve

Ako računar ne proradi „iz prve", ne dopustite da vas obuzme panika prolazne teškoće su sastavni deo amaterskog rada. Ako slika postoji ali je nestabilne, pokušajte sa podešavanjem vertikalne i horizontalne sinhronizacije TV prijemnika ili monitore (regulatori se nalaze na zadnjoj strani aparata, kod nekih televizora moraju da se podešavaju odvrtkom). Ako se na ekranu ništa ne vidi, pojačajte osvetljenje ekrana. Možda se sada, umesto jedne, vidi devet malih slika (u tri reda po tri) koje su cmo olvičene i bez teksta. Ovu pojavu nije teško otkloniti kvarc umesto na 6,144 MHz, osciluje na tri puta višoj frekvenciji. Dovoljno je da ugradite kondenzator C5 čija kapacitivnost iznosi između 10 i 30 pF. Za njegovo dodavanje, kao i za bilo koju drugu prepravku, treba isključiti računar iz mreže

Ako je računar potpuno nem, dodirnite oprezno prstom svaku komponentu, posebno IC koja. Hladnjak stabilizatora bi već posle nekoliko minuta rada trebao da bude topeo, a nešto malo i ispravljačka diode i mrežni transformator. Od čipove sme umereno da se zagreva mikroprocesor (ne toliko da ne možemo da držimo prst na njemu!) i EPROM-i. Ako je nešto pregrejano, bar znamo gde da tražimo kratak spoj

Skriveni kvarovi i ćudljive greška

Moguće je, naravno, da je kvar tako dobro „sakriven" da se još uvek nije pokazao. U tom slučaju je sasvim moguće da na štampi postoji neki kratak spoj. Isključite ispravljač, uzmite AVO-metar i na opsegu od om×1 strpljivo ispitajte sve bliska vodove. Usput proverite i da li je nožica nekog čipe ostala, možda, nezalemljena,

Štampano kolo logičke sonde

Montažna shema logičke sonde

Električna shema logičke sonde

SPECIFIKACIJA DELOVA ZA LOGIČKU SONDU

OTPORNICI

R1	390 OMA
R2	390 OMA
R3	390 OMA
R4	390 OMA
R5	390 OMA
R6	390 OMA
R7	100 OMA

KONDENZATOR

C1 100 nF

DIODE

L1-L6 LED svetleća dioda (6 KOM)

INTEGRISANA KOLA

74 LS 04
74 LS 93

a zatim okrenite štampanu ploču i ponovo proverite ispravnost rasporeda komponenti

Postoji i mogućnost da računar radi, ali uz neke specifične nedostatke kada, recimo, pritisnete neki taster pojave se dva slova umesto jednog. U tom slučaju je sasvim sigurno nastupio kratak spoj na linijama od čipova 74LS251 i 74LS156 (nalaze se jedan pored drugoga) do tastature. Ako snimite situaciju i zaključite koji se parovi slova pojavljuju zajedno, moći ćete, gledajući razmeštaj tastera u matrici (na shemi) da tačno utvrdite koje su linije kratko spojene

Moguće je da se redovi teksta na ekranu krive po horizontali naročito u poslednjim redovima. To govori o neprilagođenosti signale za sinhronizaciju slike, pa će biti neophodno da eksperimentišete sa promenom otpornosti R9 i R10 (R9 ne sme da bude manja od 40 oma, jer će u protivnom biti ugrožen čip 74LS38)

Alatka za tvrdokorne greška

Za posebno „tvrdokorne" greške treba napraviti jednu pomoćnu alatku zove se logička sonda i može biti korisna i u mnogim drugim prilikama. Za nju su potrebna dva čipa 74LSO4 i 74LS90, šest led dioda, jedan kondentator i nekoliko otpornika. Pomoću ove sonde možemo da utvrdimo da li je logički nivo na nekoj od linija visok (svetli prvi LED), nizak (drugi LED) ili postoje povorke impulsa (tada preostale četiri LED ne prikazuju statičnu situaciju nego trepere, najčešće tako brzo da imamo utisak da sva četiri svetle. statična situacija, bez povorke impulsa, ne može nikada da upali sva četiri LE diode). Najbolje je da masa i napajenje sonde budu dve raznobojne fleksibilne žice dužine oko 50 cm koje se završavaju „krokodil-hvataljkama". Njima ćemo, negde sa uređaja koji ispitujemo (to ne mora da bude samo računar „galaksija"), dovesti stabilisanih 5 V pazeći na polaritet — greška može da oštati sondu. Zatim ćemo, dodirujući zašiljenim vrhom sonde karakteristične tačke, očitavati logička stanja

Najpre ćemo se uveriti da li oscilator radi. Nožica 10 čipa 74LS32 mora da pokazuje naizmenični signal, što znači da su svi LED-ovi upaljeni. Dalje pratimo lanac delitelja nožice 2 kola 74LS93, nožica 14 kola CD4040, nožica 2 kola CD4017. Svako od ovih mesta pokazuje isto stanja na sondi, izuzev poslednjeg, kod koga je učestanost dovoljno niska da primetimo kako neki LED-ovi trepere. Ako negde postoji statično stanje, našli smo grešku. Pažljivo proverimo okolnu štampu ako na njoj nema greške moraćemo da zamenimo čip

Nožica 26 mikroprocesora mora oko pola sekunde po uključivanju da pokazuje nizak logički nivo, a zatim stalno visok. Ako nije tako, proverite tranzistor vezan za tu nožicu i elektrolitski kondenzator koji spaja R5 sa +5V

Drugi možda znaju više

Ako ni posle svih ovih operacija niste pronašli grešku, moraćete da potražite pomoć nekog stručnjaka. Čini nam se da je taj put jednostavniji nego da počnete da učite elektroniku,

Postoji, najzad, i jedan problem koji se rešava čisto softverski ukoliko je slika na vašem monitoru (televizoru) pomerana previše ulevo svaki put kada uključite računar moraćete da otkucate BYTE 11176, 12 i pritisnete **[RET]** (u ekstremnijem slučaju upotrebite naredbu BYTE 11176,13). Slično tome, ako je slika pomerana udesno, možete da otkucate BYTE 11176,10 (ili čak BYTE 11176,9) i pritisnete **[RET]** svaki put kada uključite računar

Tekst Voja Antonić Crteži Mirjana Antonić Fotografije Ivan Ivanov

Nabavka delova za računar „galaksija"

komponente i kako ih steći

Samogradnja računara, čak i u sredinama u kojima se mikroprocesori kupuju „na kilo", nije baš sasvim jednostavna stvar. Neki ključni delovi računara, kao što je ROM, ne nalaze se u slobodnoj prodaji nigde u svetu, a do nekih, kao što je tastatura, ne dolazi se ni jeftino ni lako. Kod nas, gde je često teško naći i najobičniji otpornik, upuštanje u jednu takvu avanturu može izgledati potpuno bezumno. Pokazuje se, međutim, da je moguće savladati i jednu takvu prepreku. Kako?

Zahvaljujući razumevanju i ljubavi prema računarima nekolicine domaćih proizvođača, „Galaksija" je uspela da za čitaoce ovog izdanja obezbedi barem one komponente bez kojih bi samogradnja računara predstavljala zaista samoubilački čin — ROM, tastaturu i pločicu sa štampanim vezama — i to po cenama koje su znatno ispod tržišnih! (Štampano kolo će hobiste koštati 40 odsto jeftinije nego „Elektroniku inženjering", mada oni plaćaju porez na promet, a privredna organizacija ne!) Pored toga, uspeli smo da sklopimo i dosta povoljan aranžman za nabavku poluprovodničkih komponenti iz inostranstva. U ovom času su pod znakom pitanja samo kutije računara i demonstraciona kaseta.

Mehaničke komponente

Mehaničke komponente računara „Galaksije" — štampano kolo, konektorska pločica, maska za tastere i tasteri sa kapicama — obezbeđuju Institut za vakuumsku tehniku iz Ljubljane (tasteri) i firma MIPRO i Elektronika iz Buja (sve ostalo). Tasteri koji će biti ugrađeni u računar „galaksija" zadovoljavaju sve profesionalne standarde — isti takvi se ugrađuju i u terminale nekoliko domaćih kompjuterskih sistema. Štampano kolo (razume se, od vitroplasta!) ima, takođe, profesionalni izgled i kvalitet. Vodovi su zaštićeni najpre galvanskim putem a zatim i tzv. stop-lakom (to je ona zelena boja kojoj profesionalne ploče najviše duguju za svoj šarm). Sa gornje strane je štampan raspored elemenata. Ovakav kvalitet znatno olakšava sklapanje računara, mogućnost da se neka komponenta pogrešno postavi ili da se na vodovima nepažnjom napravi „tinolski" most svedena je na teorijski minimum.

Cena kompleta iznosi 5600 dinara i određena je tako da se pokriju proizvodni i poštanski troškovi, kao i porez na promet, na koji odlazi gotovo trećina sume! (U cenu nije uračunata konektorska pločica — očekuje se da neće biti skuplja od 300 din.) Ovako popularna cena predstavlja podršku firmi MIPRO i Elektronika iz Buja i njihovih vlasnika Zvonka Jurasa i Blaža Krakića akciji „Galaksije" u širenju ideje o kućnim računarima.

Integrisana kola

Potencijalne graditelje „galaksije" ništa, valjda, ne brine toliko kao nabavka integrisanih kola. Ona se, na žalost, mogu kupiti samo u inostranstvu. Razloga za brigu ima zaista dosta: kako uskladiti narudžbu sa strogim carinskim propisima, kako objasniti na nepoznatom jeziku šta vam je, zapravo, potrebno, kako izvršiti uplatu? Postupak je, u osnovi, jednostavan: treba pisati stranoj firmi i zamoliti za profakturu. Kada predračun stigne, sa njim se odlazi u banku da bi se izvršila uplata — tzv. devizna doznaka za inostranstvo. Svako, međutim, ko je njime prošao zna koliko je težak taj put. Drugog, na žalost, nema. Jedno nikada ne gubite iz vida: maksimalna vrednost jedne pošiljke ne sme da prelazi 1500 dinara, inače će biti vraćena i nikada neće stići do vas.

Da bi bar malo pojednostavila proceduru, „Galaksija" je sklopila aranžman sa firmom „Microtechnica" iz Graca. Cena kompleta integrisanih kola, RF modulatora, kverca i tri podnožja iznosi 1660 šilinga. U cenu su uračunati i poštanski troškovi. Isporuka će biti vršena potpuno u skladu sa našim carinskim propisima. Da bi se izvršila narudžbina, dovoljno je zatražiti (na srpskohrvatskom) predračun delova za računar „galaksija". Plaćanje se može izvršiti i jednom od sledećih kreditnih kartica: American Express, Diners, Eurocard i Visa.

Svim kupcima kompleta čipova za računar „galaksija" „Microtechnica" će besplatno programirati EPROM-e. To značajno skraćuje proceduru i ubrzava put do računara „galaksija".

Narudžbinu treba izvršiti na adresu: „MICROTECHNICA", A-8042 GRAZ, St. PETER HAUPTSTRASSE 10, AUSTRIJA.

Programiranje EPROM-a

Bez sistemskih programa koje treba upisati u EPROM-e 2732 (ROM) i 2716 (karakter-generator) računar „galaksija" je potpuno bespomoćan. Čitaoci koji naruče komplet delova od „Microtechnica" dobiće isprogramirane EPROM-e — dakle potpuno spremne za ugradnju.

Čitaoci koji su već nabavili EPROM-e ili nameravaju da ih nabave preko nekog drugog distributera, treba da ih pre ugradnje pošalju redakciji na programiranje. Usluga je potpuno besplatna, a obaviće je beogradska firma MIPRO (nije greška — postoje dve firme MIPRO i obe učestvuju u našoj akciji!), u kojoj je započet razvoj računara „galaksija". EPROM-e možete početi da šaljete odmah — biće vam vraćeni u roku od petnaest dana. U pošiljku ubacite dovoljno poštanskih maraka za povratno pismo — isto onoliko koliko ste morali da zalepite na nju da biste nam je poslali. Raspitajte se, dakle, pre slanja o tarifi na svojoj pošti. Vrednosno pismo predstavlja najsigurniji način da EPROM-i stignu bezbedno do redakcije i do vas nazad.

EPROM-e treba slati na adresu: „Galaksija", 11000 Beograd, Bulevar vojvode Mišića 17.

Hitna pomoć

Neiskusni konstruktori ne treba da se plaše da će ostati sami ako negde zapnu u toku sklapanja računara „galaksija". U saradnji sa radio-klubom „Avala" iz Beograda organizovali smo službu hitne pomoći koja će dežurati svakoga dana od 17 do 20 časova uz telefon 011/402-687. Sa ovim klubom ćemo, takođe, organizovati i besplatne kurseve za sklapanje računara. Detaljnija obaveštenja ćete naći u februarskoj „Galaksiji" — u svakom slučaju pre nego što vam pođe za rukom da kompletirate delove.

NARUDŽBENICA

Ovim neopozivo naručujem komplet delova za računar „galaksija" (54 tastera, kapice sa odgovarajućim oznakama, aluminijumska maska za tastere i štampano kolo) po ceni od 5600 dinara.

Isplatu ću izvršiti poštaru prilikom preuzimanja pošiljke

____

Ime i prezime

____

L. k. i od koga je izdata

____

Ulica i broj

____

Poštanski broj i mesto

Narudžbenicu poslati na adresu: „Galaksija" — BIGZ, 11000 Beograd, Bulevar vojvode Mišića 17

Šerijska proizvodnja računara

„galaksija" u školi

Semogradnja „galaksija" predstavlja najbrži, najjeftiniji, najuzbudljiviji, ali ne i najelegantniji način da se dode do kućnog računara. Kada bi ideja o kućnim računarima zavisila samo od onih koji mogu sami da ih naprave, sa njom se ne bi odmaklo dalje od početka. Računar „galaksija", srećom, ima i svoju fabričku sestru-bliznakinju „Elaktronka Inžanjering" Iz Zemuna i Zevod za učila i nestevne sredstva privode kraju priprema za sarijsku proizvodnju komercijalna „galaksija" Ona će, prvenstveno biti namenjana školama, ali će do nje vrlo brzo moći da dođu i pojedinci.



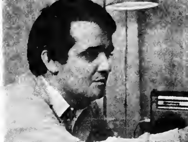
Čitaoci ovog specijalnog izdanja su, nadamo se, uvideli da će računeri u bliskoj budućnosti postati neophodni inventar svakog doma, kao što su to sada televizor ili gramofon U mnogim profesijama posao već danas ne može ni da se zamisli bez kompjutera, pa je neobično važno da svi oni koji se danas školuju i koji će u prvim decenijama 21. veka predstavljati najveći deo našeg naučno-tehničkog kadra budu što bolje upoznati sa ovom oblašću i osposobljeni da koriste računare za rešavanje problema iz svoje prakse

U začaranom krugu

Obrazovanje kompjuterskih stručnjaka je nezamislivo bez kompjutera na kojima oni mogu da proveravaju i unapređuju stečeno teorijsko znanje Na žalost, računarski sistemi kojima su opremljeni neki naši kolektivi su toliko stari da jedva obavljaju i redovni posao pa nema ni govora o tome da se njihova vrata otvora mladima u potrebnoj meri Fakulteti su najčešće opremljeni još zastarelijim kompjuterima, a škola čak ni njima Rešenja problema ovog tipa predstavlje lični računar To je relativno mali i jeftin kompjuter velikih mogućnosti koji je namenjan pojedincima U stalnom kontaktu sa njim može da se nauči programiranje reši mnogo praktičnih problema i, što je najvažnije, navikne na kompjutere i njihove mogućnosti i ograničenja Onaj ko je dobro savladao programski jezik koji njegov lični kompjuter „razume" će, bez ikakvih problema savladati i druge programske jezike i u najkraćem vremenu biti sposoban da radi sa bilo kojim drugim kompjuterom

Interesovanje za lične računare je, bar prema podacima kojima raspolažemo, ogromno i mnogi su spremni da ulože novac u kupovinu ovakve inteligentne naprave Na žalost, nadležni nemaju nimalo sluha za sve izložene činjenice Iako je ponuda računara na svetskom tržištu ogromna, jednom linearnom merom je potpuno onemogućen njihov legalan uvoz čak i kada se radi o modelima koji koštaju u današnje vreme zanemarljivih 6000 dinara Čak ni uz plaćanja visoke carine, dakle, potencijalni vlasnik računara nema mogućnosti da dođe do onoga što mu je potrebno da bi proširio svoja znanje

Jedina nada su domaće firme koje se bave sklapanjem stranih



Dva originelna domaća računara. Konstruktor „galaksije" Voja Antonić

ili proizvodnjom originalnih računara. Takvih firmi imamo dosta ali su sve na žalost, neefikasne na sajmovima se pojavljuju računari domaćeg dizajna, ali ni jedan od njih ne stiže u prodavnice. Razloge nije teško pogoditi: računari se sastoje i od integrisanih kola koja se kod nas ne proizvode, iako su ova komponente, za ono što pružaju, veoma jeftine, nabavka deviza i druge formalnosti oko uvoza su toliko složene da je praktično nemoguće proizvesti veliku seriju računara. Osim toga, domaći računari, zbog kursa dinara, koštaju toliko da ih malo ko može sebi da priušti. Škole i druge ustanove bi, doduše, mogle da odvoje 100 ili 200 hiljada dinara za kupovinu jednog ličnog računara, ali se verovatno plaše da će ta naprava biti potpuno neiskorišćena, dok se na zapadnom tržištu mogu naći desetine programa koji obavljaju čak i najspecifičnije poslove (da ne pominjemo broj programa koji mogu da se koriste u obrazovanju), programska podrška je na periferiji interesovanja naših proizvođače. Kako škole, koje nema kadar sposoban da sastavi dobre i dovoljno složene obrazovne programe, može da očekuje da će učenici videti neku korist od računara? Uz to, jasno je da preskup kompjuter neće moći da nabavi toliki broj korisnika da se na tržištu spontano pojavi veliki broj programa za njega, a nedostatak tih programa, dalje, smanjuje broj kupaca. Na prvi pogled, iz ovog začaranog kruga nema izlaza.

Golub na grani . . .

Izlaz, kako nam se čini, ipak postoji i beogradska firma „Elektronika Inženjering" je na najboljem putu da ga pronađe. Ova firma se već više od tri godine, pored ostalog, bavi i projektovanjem i proizvodnjom računara. Tako je razvijen model čiji je radni naziv EL 82: radi se o računaru velikih mogućnosti koji je zasnovan na mikroprocesoru Z80A i opremljen korisničkom memorijom od 16 kilobajta (može da se proširi do čitavih 64 Kb) i ROM-om od 16 Kb. U ROM je, pored izvanrednog bejzika koga karakteriše mogućnost operisanja sa brojevima u dvostrukoj tačnosti (15 tačnih cifara), kompletan rad sa alfanumericima, generisanje tonova i grafika solidne rezolucije, upisan i program za rad sa mašinskim jezikom koji je namenjen iskusnijim korisnicima. Računar je koncipiran tako da može neograničeno da se proširuje: umesto bejzika može da se razvije neki drugi programski jezik za specijalizovane potrebe, preko porta može da se priključi štampač, neki konvertor ili kontroler i, ako bude razvijen program koji bi to podržao (postoje linkovi koji omogućavaju da se taj program koristi bez promene ROM-a), disk-jedinice.

Kada je 1982. godine ovaj računar poprimio svoj konačni oblik i pojavio se na mnogim sajmovima širom Jugoslavije (ovaj događaj je „Galaksija" već obeležila), planirano je da maloprodajna cena bude oko 100 000 dinara. Pre nego što su rešeni problemi vezani za serijsku proizvodnju, inflacija i realan kurs dinara su učinili svoje: računar EL 82 se danas ugovorno proizvodi, ali samo za one radne organizacije koje za kompjuter u osnovnoj verziji mogu da izdvoje nekih 250 000 dinara. „Elektronika Inženjering" je procenjujući tržište, zaključila da računar koji ovoliko košta, bez obzira na povoljne karakteristike, nema šansi za veliki komercijalni uspeh.

. . . i vrabac u ruci

Zato je u saradnji sa Zavodom za učila i nastavna sredstva doneta odluka da se finansira razvoj jednog novog kompjutera. Nije vredelo ponovo početi sa prevelikim ambicijama: bio je potreban računar skromnih ali ipak zadovoljavajućih karakteristika koji će, uz to, biti toliko jeftin da će svako ko je zainteresovan biti u stanju da ga nabavi uz razumno ulaganje. Voja Antonić (koji je, uzgred rečeno, konstruktor kompjutera i EL 82) je, primenivši mnogo izvanrednih i originalnih rešenja, u potpunosti zadovoljio ovakve koncepcijske zahteve i tako je nastao računar „galaksije".

O koncepciji računara „galaksija" smo dosta pisali na stranicama ovog specijalnog izdanja obzirom da ga prezentiramo i kao projekat za samogradnju. „Elektronika Inženjering" je, naime, u želji da u Jugoslaviji bude što više vlasnika računara „galaksije" omogućila čitaocima ovog specijalnog izdanja koji poseduju izvesno praktično poznavanje elektronike da, uz minimalna novčana ulaganja, naprave ovaj lični kompjuter.

Prva karika

Nakon prvog obaveštenja u „Galaksiji" Elektronika Inženjering je dobila dosta pisama od ljubitelja računara koji, poznavajući sve probleme oko uvoza komponenti, kursnih razlika i inflacije, žele da nabave komercijalni model „galaksije" u pretplati. Komplet se sastoji od računara sa 4 K RAM-a, ispravljača, uputstva za upotrebu i priključnih kablova i košta 29.950 dinara. Za primerke iz prve serije proizvođač garantuje ovu cenu, s tim što svim kupcima odobrava kamatu od 7,5% sa očekivanim vremenom isporuke u martu 1984.

Kada računar „galaksija", putem kupovine ili kao samogradnja, postane vlasništvo većeg broja Jugoslovena, pojaviće se značajan broj programa za njega, koji će u mnogome povećati vrednost ovog kompjutera. Ti programi će se koncentrisati oko časopisa za popularizaciju nauke „Galaksija" koji će na svojim stranicama odvojiti značajan prostor za međusobne kontakte vlasnika računara. **Svi** proizvedeni modeli računara i svi modeli koje čitaoci ovog specijalnog izdanja naprave biće potpuno jednaki tako da će za njih moći da se koriste isti programi bez ikakvih prepravki.

Veliki broj vlasnika, ipak, mora da postavlja i velike zahteve: bazična verzija računara je skromna pa su predviđena proširenja. „Elektronika Inženjering" će proizvesti hardverske dodatke koji su primereni mogućnostima i potrebama vlasnika računara „galaksije". To će verovatno biti moduli za proširenje memorije, palice za igru, generator tona, AD i DA konvertori ... U samoj kutiji računara je, čak, predviđeno mesto za EPROM u koji će jednog dana biti upisan program za rad sa mašinskim jezikom, prošireni bejzik, program za obradu teksta ili nešto drugo što će biti diktirano potrebama vlasnika kompjutera, koje će se tek ispoljiti. Sve u svemu, nijedan potencijalni vlasnik računara neće biti prinuđen da kupi nešto što mu nije potrebno samo zato što je to uključeno u bazičnu verziju računara, ali će svako, u toku sledeće godine, moći da dođe do onoga što može da mu pomogne u poslu kojim se bavi. Što se dizajniranja i proizvodnje periferijske opreme tiče, Elektronika Inženjering poziva sve zainteresovane, naročito iz „male privrede", na konkretnu saradnju, koja može da se ostvari kroz razne ugovorne oblike ili udruživanjem. Pišite im — sigurni smo da ćete pronaći zajednički jezik. Svaka ideja u interesu vaše i naše „galaksije" je dobrodošla.

Čini nam se da je računar „galaksije" odnosom mogućnosti/cene konkurentan mnogim stranim komercijalnim modelima: da kod nas ima mogućnosti za proizvodnju ogromnih serija, mogli bi da se projektuju specijalizovani čipovi (poput čuvenog Sinclairovog ULA čipa) koji bi zamenili značajan deo njegovog hardvera i pojednostavili konstrukciju, smanjujući samim tim i cenu. Zato se „Elektronika Inženjering" zalaže za daleko fleksibilniji stav nadležnih prema uvozu računara (ma koliko ja sadašnji propisi štite): što kod nas bude bilo više kompjutera i mladih ljudi koji su osposobljeni da se njima bave, to će računar „galaksija" i neki njegov eventualni naslednik (o kome se, trenutno, još ne razmišlja ali koji je neizbežan s obzirom na neprekidni napredak elektronike), naići na šira i spremnija tržišta.

Adresa „Elektronike Inženjering" je Karađorđev trg 11, 11070 Zemun, a za sva obaveštenja možete da se obratite i na telefone 601 669 i 601 577.

7 male tajne velikih majstora

Kada savladamo osnove programiranja, poželećemo da koristimo računar za rešavanje problema sa kojima se srećemo u svakodnevnom životu. Većina korisnika, ipak, nema dovoljno takvih problema da bi skupo plaćeni računar bio u dovoljnoj meri „zaposlen" na njihovom rešavanju, pa im ostaje da računar zaposle i drugim poslovima. Kupovanja komercijalnih programa za igre (a o takvim programima je bilo dosta reči u ovom izdanju) pruža prilično zabave, ali zabave koja je po svojoj prirodi nekreativna. Pisanja programa za igre je, bar za neke, mnogo zanimljivije, ali je za pisanje igara potrebno mnogo različitih i originalnih ideja čiji izvor, pre ili kasnije, presuši. Jedna grupa korisnika se tada okreće hardverskim proširenjima računara. Kako na računar može, na primer, da se priključi dobar sintetizator govora? Može li, pored njega, da se priključi i analizator zvuka koji bi omogućio da računar, naravno ograničeno, razume govor i odgovara na njega? Izrada hardvera koji ovo omogućava je vrlo složen posao, a pisanje programa koji bi kontrolisao taj hardver još složeniji.

Kako prevariti računar

Sigurno je da se dogradnjama računara možemo zabavljati čitavog života i da nam rad na ovom planu može donati mnogo prijatnih trenutaka. Ipak, za ovakve poslove je neophodno solidno poznavanje elektronike, dosta iskustva u samogradnjama (mada se, jasno, iskustvo na može steći bez samogradnji) i određena materijalna sredstva za nabavku komponanti. Sigurno je da dobar deo vlasnika računara na poseduje sve to. Ipak, i oni mogu da se zabavljaju „proširujući" svoj računar, a tim što to proširivanje neće biti hardversko nego programsko.

Operativni sistem računara je vrlo složen program koji bi trebao da reaguje ispravno u svim slučajevima i da, kada korisnik preduzme neku akciju koja nije predviđena, prijavi odgovarajuću grešku. Jasno je, međutim, da konstruktor računara i programer koji je pripremao operativni sistem na mogu da predvide svaku akciju korisnika. Računar, ipak, na svaku akciju mora nekako da odgovori i tako nastaje ono što se u programerskom žargonu naziva BAG (bug na engleskom znači bubica, ali se ovde odnosi na slučaj u kome računar (ili neki program) na radi onako kako je predviđeno). U većini slučajeva „bag" na donosi ništa korisno — ukoliko korisnik nepravilno rukuje računarom, vrlo je verovatno da će se on blokirati i neće izvršavati nikakve naredbe sve dok mu se memorija na obriše isključivanjem iz mreže. Ipak, ponekad „bag" može da omogući pristup nekim internim operacijama računara i pruži korisniku mogućnost da, u većoj ili manjoj meri, prilagodi operativni sistem računara nekim svojim potrebama. Pažljivo proučavanje i klasifikovanje „bagova" predstavlja softversku dogradnju računara i veliki korak ka potpunom razumevanju njegovog rada.



Traganja za nestandardnim tehnikama i instrukcijama je posebno okupilo jednu grupu korisnika koja je izgubila interes za bilo kakvu drugu upotrebu računara. Kako piše eminantni časopis „Byte", postoje vlasnici računara koji najveće zadovoljstvo

Ovo poglavlje namenjeno je vlasnicima stonih i džepnih računara koji su savladali osnovne korake u programiranju i zaželeli da iskoriste svoj računar do maksimuma. Posle nekoliko uvodnih napomena, opisaćemo neke manje poznate karakteristike pojedinih računara i predložiti neka područja njihove primene. Istraživanja manje poznatih tehnika zahtevaju „beskrajno“ strpljenje i okupiraju svo slobodno vreme zainteresovanog vlasnika računara. Većina nestandardnih postupaka, ma koliko vreme bilo utrošeno na njihovo pronalaženje, ostaje bez praktične vrednosti i primene. Ipak, istraživanja ovog tipa su dala i mnogobrojne korisne rezultate, od kojih će neki ovde biti izloženi.

programiranja

nalaze u tome da pritiskaju po pet tastera istovremeno i posmatraju šta će da se desi ili izvršavaju programske sekvence lišene svake logike u nadi da će „prevariti" računar. Takvi korisnici, piše „Byte", ne žele da pišu programe niti da pronalaze bilo kakvu aplikaciju onoga što su pronašli; njihov cilj je samo pronalaženje.

Rizik bez rizika

Ovakvi ljubitelji računara su, očigledno, zaboravili da računar, u osnovi, služi za rešavanje praktičnih problema, ali im se mora priznati neverovatna upornost i značajna inteligencija. Iako je dobro što većina nas nije u njihovoj koži, čini nam se da svako treba da se upozna sa korisnim nestandardnim instrukcijama i tehnikama koje mogu da se primenjuju na njegovom računaru. Nema nikakvog opravdanja za one koji se plaše korišćenja bilo čega što nije jasno opisano u okviru uputstva za upotrebu računara — korišćenje nestandardnih tehnika može da bude neobično korisno, a njihovo razumevanje zanimljivo.

Pre nego što pređemo na konkretne primere, treba da kažemo da gotovo ni jedan proizvođač računara ne podržava primenu nestandardnih tehnika. Razlog za to nije u tome što su one štetne već u tome što izazivaju često nepredvidljivo ponašanje računara. Gubici memorije, „blokiranja" tastature, brisanje nekih delova memorije i prikazivanje „nepredviđenih" karaktera na ekranu kose se sa ambicijama svakog proizvođača da njegov računar deluje tačno onako kako je propisano. Proizvođač obično nema ni dovoljno vremena ni dovoljno kadra da proučava (kako se izrazila firma Hewlett-Packard") „rudimente operativnog sistema računara". Ekipa koja je konstruisala neki računar se najčešće raspada posle završetka projekta, jedan deo ljudstva biva angažovan na drugim zadacima, a neki i menjaju firmu u kojoj su zaposleni. Dakle, nema ko da odgovara na korisnička pitanja ovog tipa, pa ova jednostavno bivaju ignorisana. Zato američki stručni časopisi sve nestandardne tehnike klasifikuju terminom NOMAS (**No Manufacturer Support** tj. ono što proizvođač ne podržava). Članci označeni ovakvim zaglavljem upozoravaju korisnika da ne pita proizvođača računara o onome što mu nije jasno u vezi onoga što je u tekstu spomenuto. Treba se obratiti njegovom autoru ili, jednostavno, „mućnuti glavom". U smislu te klasifikacije, dalji deo ovog poglavlja može da bude označen sa NOMAS.

maternji jezik računara

Većina stonih računara, kako često kažemo, radi na bejziku. Ova tvrdnja je samo delimično tačna: stoni računari, sa gledišta korisnika, rade na bejziku jer on kuca naredbe na tom jeziku i na njemu dobija odgovore od računara. Sam računar ne „razume" bejzik — u njegov ROM je ugrađen vrlo složen program koji naredbe bejzika prevodi na za računar jedino razumljiv jezik — mašinski jezik ili mikrokod — koji se sastoji od velikih nizova nula i jedinica. Maternji jezik računara se, dakle, ne odlikuje velikom konciznošću i preglednošću, (o razumljivosti da i ne govorimo), a ipak je bolji od bilo kog postojećeg programskog jezika!

Čitavo ovo poglavlje posvećeno je onima koji, poznavajući blagodati bejzika, žele da se upuste u pisanje mašinskih programa. Zar ima takvih? Ima ih, i to mnogo — svaki ozbiljan vlasnik računara će, pre ili posle, preći na mašinski jezik. Zato, pre nego što pređemo na ovaj jezik, moramo da objasnimo potrebu za njim.

Razgovor bez prevodioca

Pre svega, zamislite da razgovarate sa nekim Francuzom neznajući njegov jezik. Razgovor se, dakle, odvija preko prevodioca. Očigledno je da je potrebno vreme da ono što ste želeli da kažete objasnite prevodiocu, da zatim on prevede vašu rečenicu, saopšti je Francuzu, razume njegov odgovor i prevede je vama. Isto je tako i kod računara — prevođenje svake naredbe traje dugo. Ako se jedan ciklus u programu izvršava 1000 puta, naredbe unutar tog ciklusa će 1000 puta biti prevođene. Programi koji rade na mašinskom jeziku, zato, po pravilu rade 40—100 puta brže od odgovarajućih BASIC programa.

Problemi sa brzinom će, u daljem razvoju računara, verovatno biti prevaziđeni, ali potrebe za mašinskim jezikom neće. Može te li Francuzu sa kojim razgovarate posredstvom prevodioca da saopštite **tačno** ono što mislite? Njemu ne govorite vi nego prevodilac, on nehotice modifikuje neke vaše rečenice tako da se nijansa njihovog značenja može lako promeniti. Bejzik omogućava realizaciju čak i vrlo složenih programa za obradu podataka, ali kada se dođe do osetljive oblasti rada sa periferijskim jedinicama (štampačem, ploterom, modemom i, naročito, disk-jedinicom) počinju da se javljaju „nesporazumi" — korisnik tačno zna šta bi računar trebalo da uradi, a računar radi nešto slično tome, ali ne to! Korisnik tada, u većini slučajeva, modifikuje svoje potrebe, što izaziva delimični gubitak poverenja u računar.

I ovaj problem može da bude prevaziđen bez mašinskog jezika — postoje specijalizovani viši programski jezici za rad sa periferijskom opremom. No, vratimo se opet razgovoru sa Francuzom i prevodiocu: šta ako želite da razgovarate o nekoj specifičnoj temi (npr. računaru ZX Spectrum) koju i vi i sagovornik odlično poznajete ali prevodilac o njoj nema pojma? Proći će mnogo, mnogo vremena dok prevodilac ne shvati ono što želite da kažete, njegov prevod će, u nedostatku poznavanja stručnih termina, postati nerazumljiv i na kraju ćete morati da prekinete razgovor. **Nikada ni jedan** konstruktor računara i programskih jezika neće moći da predvidi sve potrebe korisnika i da obuhvati sve oblasti u kojima će računar biti korišćen. Mašinski jezik omogućava korisniku da natera računar da radi tačno ono što njemu treba bez posredstva operativnih sistema. Pomoću mašinskog jezika realizuje se složeni softver za obavljanje specifičnih potreba koji, docnije, lako može da preraste u poseban programski jezik.

Treba, ipak, reći da većina prosečnih korisnika računara počinje da piše mašinske programe ne zbog zadovoljavanja potreba za nekim sasvim specifičnim aplikacijama računara već da bi programi za igre brže radili. Ovo se posebno odnosi na vlasnike sporog ZX81.

Mašinska abeceda

Iako su mašinski jezici, kao što smo rekli, nizovi jedinica i nula, oni se prikazuju na nešto pogodniji način preko mnemoničkih skraćenica. Pre svega, jedinice i nule treba pretvoriti u heksadekadne brojeve (smatramo da vam je poznato šta su heksadekadni brojevi — u njima se, osim cifara 0—9, javljaju i slova A, B, C, D, E, i F koja redom predstavljaju „cifre" 10, 11, 12, 13

14 i 15). Zatim se heksadekadni brojevi grupišu dva po dva (jer su dva heksadekadne cifre jedan bajt dok se, sa stanovišta memorije, jedna heksadekadna cifra naziva „nibble". Zatim se programske instrukcije zamenjuju skraćenicama koja opisuju njihovo dejstvo. Kod procesora Z80 se, na primer, često koristi naredba LD A, H koja se heksadekadno kodira kao 7C. Skraćenica LD dolazi od „load" i znači: prenesi sadržaj registra H u registar A (akumulator).

Za pisanja programa na mašinskom jeziku potrebno je, najpre, shvatiti strukturu naredbi. Svaka naredba se sastoji od mnemoničke skraćenice (u gornjem primeru to je bilo LD) koja se prevodi u dve ili četiri heksadekadne cifre (jedan ili dva bajta). Iza skraćenica se piše adresni deo koji govori računaru gde se nalaze podaci koje traba da obraduje. Ovaj adresni deo može da bude neki od procesorskih registara (B, C, D, A, E, H, L kod procesora Z80, na primer) ili adresa nekog bajta memorije. Naredba LD A, (7530), na primer, prenosi sadržaj memorijskog bajta čija je adresa 7530 (ova adresa je heksadekadna i njoj odgovara dekadna adresa 30000) u procesorski registar A radi obrade.

Kada smo napisali program u mnemoničkim skraćenicama, pristupamo njihovom pretvaranju u heksadekadne brojeve. Za to možemo da koristimo tabelu (prema njoj se gore pomenuta naredba LD A, (7530) prevodi sa 3A 30 75) ili gotov program koji vrši prevođenje. Ovakav program može da bude više ili manje složen, a vrlo su popularni složeni „asemblar programi" koji omogućavaju unošenja skraćenica, ispravljanja, prevođenja programa i snimanja na kasetu. Ukoliko radimo manuelno, treba dobijene heksadekadne brojeve pretvoriti u dekadne (za to se koristi bejzik program ili, najčešće, neki programabilni džepni računar) i uneti ih u memoriju pomoću naredbe POKE.

Da sve ovo ne bi ostalo na nivou teoretisanja, sastavićemo kratak mašinski program koji uvećava sadržaj nekog bajta memorije za 1 i odgovarajući prateći bejzik program. Mašinski program je pisan za procesor Z80 koji poseduje većine popularnih modela računara (ZX81, Spectrum, TRS 80 ...) koji su kod nas zastupljeni. Naš mašinski program će počinjati od adrese 30000 (heksadekadno 7530) i imaće svega četiri instrukcije. Prva instrukcija (LD A, (753A)) će u registar A (često se naziva i akumulator) dovesti broj koji se nalazi na adresi 753A (dekadno 30010), sledeća će povećati sadržaj registra A za jedan (INC A), zatim će sadržaj akumulatora biti vraćen na odgovarajuće mesto u memoriji (LD (753A),A) i na kraju će se izvršavanje mašinskog programa prekinuti (naredba RET) i računar će se vratiti u bejzik program iz koga je usledio poziv mašinskog. Ceo program, kodiran, izgleda ovako.

```
7530    3A 3A 75    LD A, (753A)
7533    3C          INC A
7534    32 3A 75    LD (753A),A
7537    C9          RET
```

Sa leve strane se nalaze adrese instrukcija, a odmah do njih su same instrukcije, predstavljene kao heksadekadni brojeva. Mnemoničke skraćenice su ispisane pored njih radi kompletnosti — one računaru ništa na znače.



Jezik mašine. Da biste svoj kućni računar pretvorili u poslušnu spravu, morate, pre svega, dobro naučiti njegov maternji jezik

Rutine iz ROM-a

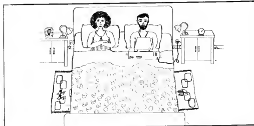
Program glasi 3A 75 3A 3C 32 75 3A C9 ili, kada ove brojeve pretvorimo u dekadne 58 117 58 60 50 117 58 201. Sada nam ostaje da sastavimo bejzik program koji će uneti mašinske instrukcije u memoriju, a zatim demonstrirati rad programa.

```
 10 FOR I=0 TO 7
 20 READ A
 30 POKE 30000+I,A
 40 NEXT I
 50 INPUT A
 60 POKE 30010,A
 70 CALL 30000
 60 PRINT PEEK 30010
 90 GOTO 50
100 DATA 58,117,58,60,50,117,58,201
```

Nije teško razumeti rad programa. Najpre se iz DATA naredba u memoriju unosi mašinski program koji počinje od adrese 30000. Zatim se od korisnika traži da otkuca neki broj (podrazumeva se da je taj broj manji od 256), a zatim računar poziva potprogram i na ekranu prikazuje rezultat.

Čak i na ovom jednostavnom primeru se vidi koliko se mašinski programi razlikuju od računara do računara. ZX81, na primer, ne poseduje naredba READ i DATA, pa bi unošenja programa moralo drukčije da se organizuje. Naredba CALL 30000 bi kod mnogih računara morala da se zameni sa, na primer, B=USR(30000), a kod nekih drugih bi broj 30000 morao da bude smešten negde u memoriji, pa bi se koristilo B=USR(0). Kod računara koji ne koriste Z80 procesor morala bi da se izvrše korenite izmene: čitav mašinski program bi, što se kodova tiče, bio sasvim različit, a možda ne bi imao ni isti broj naredbi. Kada bismo ovaj program želeli da primenimo na računaru BBC model B, **sve** instrukcije bejzik programa bi morale da budu promenjena, a neke uopšte ne bi ličila na one koje smo koristili!

Lako je primetiti da sastavljanje mašinskih programa nije ni malo jednostavan posao. Naš program je „znao" gde se nalazi argument operacije koju treba da izvrši i gde da smesti rezultat. Da smo, međutim, bili ambiciozniji i poželeli da eliminišemo potrebu za bejzik programom i



da na mašinskom jeziku realizujemo unošenje broja sa tastatura i prikazivanje rezultata na akranu, naš program bi se produžio na stotinak instrukcija. Baš u ovoma i leži umešnost dobrog programera cilj je upotrabiti rutina iz ROM-a računara za obavijanje nekih zadatakai Računar, jasno, negde u ROM-u ima potprogram koji služi za unošenja podataka sa tastature (ovaj potprogram se koristi za bejzik naredbu INPUT) Zar na bi bilo bolje pozvati ovaj program? Na žalost, ROM je toliko veliki da ovaj program nije lako pronaći, a nije lako ni zaključiti u kom će se memorijskom bajtu naiaziti broj koji je korisnik otkuoso posle njagovog izvršavanja

Kako ući u ROM?

Disasembliranja (prevođenja nekog mašinskog programa na mnemonička skraćenios i razumevanja njegovog delovanja) čitavog ROM-a je posao čija složenost prevazilazi mogućnosti 99% korisnika. Porad toga, ovom poslu je potrebno posvetiti se nekoliko meseci da bi sa okončao na zadovoljavajući način Na sraču, postoje takozvana „supermape" bejzika pojedinih računara. Njih su sastavili korisnici koji su disasenblirali ROM svog računara i koji sada prodaju razultate svoga rada ostalima ili (u retkim alučajevima) sami proizvođači računara

„Supermapu" bejzika računara TRS 80 sastavio je Amerikanac Rodžer Fuier (Roger Fuller) Na njenih četrdesetak strena naiaze se adrese svih važnijih (pa čak i većina nevažnih) rutina u ROM-u, kao i objašnjenja aistama adresiranja. Jednostavnim pogiedom u supermapu vidimo da se početak mašinakog potprograma koji realizuje naradbu iNPUT nalazi na 219A, a da se ono što korisnik otkuoa unosi u „bafer" koji se nalazi na adresama 41E8—42E8

Supermapa je, dakla, neophodna alatka za onoga ko želi da se upusti u mašinsko programiranje na ozbiijnijem nivou. Supermapu, međutim, mogu da koriste i oni koji ne žele da se beve mašinakim programiranjam već žeie da iskoriste neka „akrivene" mogućnosti računara. Korišćenja tih mogućnosti obradićemo na primeru koji se ponovo odnosi na TRS 80.

U toku izvršavanja programa pritisak na taster BREAK izaziva prekid rada i ispisivanja READY na akrenu. Zamisiimo da ssatavljamo bejzik program koji će simulirati neki drugi programski jezik (autor se sa ovim probiamom susreo u toku sastavijanja programa koji simulira zamišljeni računar NAR 1 sa kojim se srednjoškolci upoznaju u okviru predmeta računari i programiranje) U toku rada intepratatora pritisak na BREAK treba da prekine izvršavanje programa ali na programa-simulatora nego programe čiji se rad simulira! Dakle, pritisak na BREAK treba da izazove prelazak na izvršavanja neka unapred predviđena bejzik naredbe Kako to izvesti bez mašinskih programa?

Konsultacija supermape daje da se, po pritisku na taster BREAK izvršave naredbe koja je smeštena na 400C (dekadno 16396) Pomoću PEEK se uvaravamo da sa na tom mestu naiazi naredba RET (C9 ili 201 dekadno), a da au sledeća dve bajta slobodna Upravo ono što nam je potrebno! Unesimo u naš bejzik program naredbe POKE 16396, 195 POKE 16397,45 POKE 16398,1. Na ovaj način će računar, kada pritisnemo BREAK, izvršiti mašinska naredbe JP 012D čime će se izvršavanje pranati na program u ROM-u koji počinje od 012D Ovaj broj smo ponovo dobili konsultacijom supermape — tamo se nalazi rutina koju računar normaino izvršava kada naiđe na nepostojaču naredbu, u kom slučaju se na ekranu izdaje poruka „L3 ERROR" Ukoliko u okviru našeg bejzik programa dodamo naredbu ONERRORGOTO 1000, svaki pritisak na taster BREAK će izazivati prelazak na liniju 1000 Tamo, ukoliko trebe da usledi prekid reda programa, može da sa nađe naredba POKE 16396,201 čime će red tastera BREAK biti normalizovan.

Još jedan primer upotrebe supermape je realizacija novih naredbi TRS 80, na primer, ima tridesetak naredbi koje su aktivne samo kada su priključeni diskovi. Ukoilko njih nema, nije tešo sastaviti mašinske potprograma koji ove naredbe preimenuju u neke nove koje su korisniku potrebne. Na primer, kada računar naiđe na naredbu MERGE, on izvršava instrukciju koja se nalazi na adresi 418B Ukoliko tamo, pomoću POKE, postavimo mašinsku naredbu JP 7530, naredbe MERGE u programu će izazivati početak izvršavanja našeg mašinskog programa koji počinje od adrese 30000i Ova tehnika se, na žalost, na može primeniti kod sinklarovih računara jer njihovi konstrukton nisu predvideli tzv link za izlazak iz ROM-a

Prevođenje programa

Deo namenjen iskusnijim programenma koji na bi obradio priiagođavanje programa pisanih za jedan računar nakom drugom modelu na bi bio kompletan. Računari koji su dugo na tržištu kao što su Appia II+ i TRS 80 imaju moćnu programsku podršku — hiljada ijudi je napisalo desetina hiljada programa koji mogu da sa nađu u knjigama, kopiraju od prijatelja i slično Noviji računari su, po mogućnostima, daleko moćniji od atarih, ali im je programska podrška slaba — generacija računara sa u posladnje vreme smenjuju toliko brzo da nema vremena za pisenje dovoljnog broja programa Zato nije malo onih koji požele da prilagode postojeće programe svome računaru

Ako je program pisan na bejziku i ako se u njemu ne koriste naredbe PEEK, POKE pozivi mašinskih rutina iz ROM-a (naredbe CALL), on se, u principu, može prilagoditi svakom računaru. Postoje, naravno, programi koje je gotovo nemoguće preraditi — neki numerički proračun koji se zasniva na toma da TRS 80 računa sa 13 tačnih cifara na može da sa prilagodi za ZX81 koji računa se 4—5 tačnih cifara Isto tako, program koji zahteva grafiku visoka rezolucije teško može da se prilagodi računaru koji ovu rezoluciju na poseduje, a da ne bude „osakaćen"

Programe koji sadrže PEEK i POKE možete da prilagodite vašem računaru ako pronađete nekoga ko poseduje računar za koji je program pisan i ko može da vam kaže šta je tim PEEK i POKE naradbema postignuto. Ukoliko se u programu na početku nalazi neko POKE, vrlo je varovatno da ono služi za iniciranje nekog posebnog moda računara (npr broja karaktera u redu i slično) pe ćete verovatno moći potpuno da ga zanamarite Ukoliko je POKE korišćena više puta u raznim delovima programa, moraćete da shvatite zašta ona služi ili da nađete nekoga ko će vam to objasniti.

Ukoliko naiđete na program koji sadrži mašinske potprograme, dobro promislite pre nego što predate na prevođenje Ukoliko ti mašinski potprogrami makar na jednom mestu pozivaju neku rutinu iz ROM-a, savatovali bismo vam da odustanete od posla — mnogo će lakše biti da napišete nov program nego da prilagodite atari

Postoje dva tipa prevođenja programa Jedan zahteva da se uklopimo u logiku polaznog programa, shvatimo kako on radi (eventualno nacrtamo njegovu blok-shemu), a zatim napišemo novi program Drugi način je mehaničko prevođenja programa čitamo naredbu po naredbu i one naredbe koja su zajednička za oba računara pranosimo bez razmišljanja Ukoliko naiđemo na neku naredbu koju naš računar na poznaje, pokušavamo da je zamenimo naradbom (ili grupom naredbi) našeg računara koja će imati isto dejstvo. Pri radu na drugi način mi na shvatamo kako program funkcioniše već ga „kopiramo" uz rešavanja problema „u letu"

Svaki iskusniji korisnik računara će, ako je ikako moguće, težiti da prevodi program na drugi način jer sa tako štedi u vremenu Problem se javlja kada prevedemo program i vidimo da on na radi Pronalaženje grešaka je vrlo jednostavno ukoliko imamo pristup do računara za koji je program pisan — postavljanje STOP naredbi na strateški dobro izabrana mesta u programu i provaravanja sadržaja promenljivih bi trebalo da dovede do brzog iociranja i ispravijanja

Različite naredbe — isto dejstvo

	Appls	BBC	TRS 80	VIC/Pet	Spectrum	ZX81
Biranje slučajnog broja između 0 i 0.99999999	RND(1)	RND(1)	RND(0)	RND(1)	RND	RND
Biranje slučajnog broja između 1 i N (ceo broj)	RND(1)*N+1	RND(N)	RND(N)	RND*N+1	RND* N+1	RND*N+1
Biranje slučajnog slova A—Z	CHR$(INT(RND (1)*26+65))	*CHR$(RND(26) +64)	CHR $ (RND(26) +64)	kao Apple	CHR $ (INT(RND *26+65))	CHR$(INT(RND*26+38)
Brisanje ekrana	HOME	CLS	CLS	PRINTCHR$ (147)	CLS	CLS
Unošenje podatka za vreme rada programa	H$="" IF PEEK (—16384)' >127 THEN GET X$	INKEY$(N)	INKEY$	GET X$	INKEY$	INKEY$
Prevođenje karaktera u kod	ASC („X")	ASC („X")	ASC („X")	ASC („X")	CODE („X")	CODE („X")
Pomeranje kurzora gore	CALL —998	PRINT CHR$ (11)	PRINT SHR$(11) (27)	PRINT CHR$ (145)	PRINT CHR$(11)	PRINT CHR$(112)
Pomeranje kurzora dole	PRINT CHR$ (10)	PRINT CHR$ (10)	PRINT CHR$ (26)	PRINT CHR$ (17)	PRINT CHR$(10)	PRINT CHR$ (113)
Pomeranje kurzora levo	PRINT CHR$ (8)	PRINT CHR$ (8)	PRINT CHR$ (24)	PRINT CHR$ (157)	PRINT CHR$(8)	PRINT CHR$ (114)
Pomeranje kurzora desno	PRINT CHR$ (21)	PRINT CHR$ (9)	PRINT CHR$ (25)	PRINT CHR$ (29)	PRINT CHR$ (9)	PRINT CHR$ (115)
Izdvajanje N2 karaktera iz teksta počevši od N1-tog	MID$(A$ N1, N2)	MID$(A$,N1, N2)	MID$(A$,N1, N2)	MID$(A$,N1, N2)	A$(N1TO N1+N2−1)	A$ (N1 TON1+N2−1)
Izdvajanje prvih N karaktera iz teksta	LEFT$(A$ N)	LEFT$(A$ N)	LEFT$(A$ N)	LEFT$(A$ N)	A$(1 TO N)	A$(1 TO N)

greške. Ukoliko nemamo na raspolaganju oba računara, moramo se delimično uklopiti u logiku programa u nadi da ćemo naći variable koje imaju nelogične vrednosti i istražiti uzrok ovih nelogičnosti. Posedovanje štampača znatno olakšava prevođenje obzirom da svaki računar poseduje neki TRACE mod (u ovom modu se ispisuju (na ekranu ili štampaču) sadržaji promenljivih posle svake naredbe, linijski brojevi naredbi koje se izvršavaju i slično).

Sve u svemu, prevođenje složenih programa na jezike drugih računara zahteva dobro poznavanje bar dva računara, prilično strpljenje i upornost. Neki programi se lako prilagođavaju, dok je to kod drugih praktično nemoguće. Tabela pokazuje naredbe sa istim dejstvom kod nekoliko različitih računara.

Mašinac iz bejzika

Većina bejzik računara ima mogućnost programiranja na mašinskom jeziku što znači da se, regularnim naredbama, može pristupiti svim sistemskim registrima, menjati sadržaj bajtova u memoriji i slično. To se najčešće postiže BASIC naredbama PEEK, POKE i CALL (BBC računari, na primer, ima nešto drugčije naredbe, ali one imaju isto dejstvo; kod nekih računara se umesto CALL koristi OPN, a kod nekih USR). Naredba PEEK zahteva adresni deo. A=PEEK 16000 (kod nekih računara 16000 mora da bude u zagradi) će promenljivoj A dodeliti vrednost bajta koji, u apsolutnoj mapi adresa koje obuhvata ROM i RAM, ima broj 16000. Naredba POKE ima dva adresna dela razdvojena zarazom. POKE 16000,11 će u bajt čija je adresa 16000 staviti sadržaj 11 (dekadno)*

Naredba CALL (prati je adresa) započinje izvršavanje mašinskog programa čiji se početak nalazi na zadatoj adresi. Povratak u bejzik program mora da obezbedi sam mašinski potprogram odgovarajućom RTN naredbom.

Ništa lakše od smeštanja proizvoljnog sadržaja u neki bajt memorije, ali rezultati takve akcije mogu da budu vrlo neprijatni. Pre svega, postoji mogućnost da naredba POKE ne deluje. rekli smo da se RAM i ROM adresiraju zajedno; kod većine računara adrese 0—15000 predstavljaju ROM koji, jasno, ne može da se menja. Ukoliko pokušate da, uz pomoć POKE naredbe, smestite neki broj na adresu u ROM-u, ništa se neće dogoditi i imaćete utisak da je naredba pravilno izvršena. No, ako pomoću PEEK pokušate da proverite dejstvo naredbe POKE, videćete da ona uopšte nije delovala.

Može, isto tako, lako da se desi da, pomoću POKE, unesete neki broj u deo memorije koji sadrži program koji ste prethodno otkucali. Ukoliko taj broj dođe u sredinu neke bejzik naredbe, neće biti nekih velikih problema — jedna instrukcija će biti zamenjena drugom i to će, odgovarajućim editovanjem ili ponovnim kucanjem iste naredbe, lako moći da se ispravi. No, u okviru svake naredbe se, obično, nalazi i adresa sledeće. Ukoliko nehotice poremetite ovu adresu, računar će dospeti u totalnu komfuziju i potpuno će pogrešno prikazivati i izvršavati program (verovatno ćete, kada otkucate LIST, videti samo naoko besmislene karaktere kako ista po ekranu). Kucanjem naredbe NEW ovakav problem će sigurno biti korigovan.

Najveća opasnost leži u menjanju sadržaja neke sistemske promenljive. Tu se nalaze konstante neophodne za ispravan rad računara: adresa prvog korisnikovog programa, adresa kraja memorije, početna i krajnja adresa dela memorije u kome su smeštene informacije koje računar prikazuje na ekranu, adresa instrukcije koja se trenutno izvršava i slično. Ukoliko promenite neke od ovih konstanti, rezultati su nepredvidljivi. Moguće je da se neće ništa dogoditi. Moguće je da se u početku neće ništa dogoditi i da će se problemi pojaviti nekoliko časova posle primene inkriminisane POKE naredbe. No najverovatnije je da će problemi nastupiti istog momenta: program će „nestati" iz memorije ili će se pojaviti neki delovi; možda će se ekran „smanjiti" ili će računar prijaviti besmislene greške ili će se, jednostavno, sve blokirati. Ukoliko se desi poslednje, ostaje vam samo da isključite i ponovo uključite računar.

Naredba POKE je, dakle, vrlo opasna ako ne znate tačno šta radite. Da biste znali šta radite, morate da imate tzv. mapu memorije. U mapi memorije su ubeležene početne i krajnje adrese RAM-a i ROM-a i sistemska promenljive i napomenuto šta je smeštano u svaku od njih. Ovakve mape se najčešće nalaze u uputstvima za upotrebu računara ili u posebnim knjigama posvećenim programiranju na mašinskom jeziku.

ZX81 bez tajni

ZX81, svakako, spada u računare kod kojih je otkrivanje skrivenih karaktaristika imparativ za svakog ozbiljnog korisnika. Razloge nije previše teško pogoditi: mogućnosti ovog računara su relativno male, a postoje mnogobrojne dobra osobine koje su mu nepoznate. Na svu sreću, konstruktori ZX81 su u njagov operativni sistem uključili naredbe PEEK, POKE i USR koje otvaraju vrata programiranja na mašinskom jeziku i time omogučavaju korisniku da zaobiđe operativni sistem koji su konstruktori zamislili.

Ako malo eksperimentišete sa naredbom POKE, videćete da se za većinu adresa ne primećuju nikakve reakcije, ali da smeštanje brojeva na adrese 16384 — 16508 izaziva raznorazne posledice. U ovom području se, naime, nalaze takozvane „sistemske promenljive", odnosno brojevi koji govore računaru kakav se program nalazi u memoriji, koliko memorije ima na raspolaganju, koliko redova ekrana je ispunjeno, koje je greška nastupila i slično. Bezmalo, svaka sistemska promenljive zauzima dva bajta, u tom prostoru su smeštene dva broja koja odvojeno posmatrano nemaju smisla, ali koji mogu da se ujedine u jedan značajan broj.

Spuštanje i podizanje RAMTOP-a

Uzmimo, na primer, sistamsku promenljivu nazvanu RAMTOP (ime RAMTOP — vrh, kraj memorije — je samo figurativno i za ZX81 ne znači ništa, PRINT RAMTOP neće na ekranu prikazati vrednost neke sistemska promenljive) koje se nalazi u ćelijama memorije 16388 i 16389. Da bismo pronikli u njen sadržaj, otkucaćemo PRINT PEEK 16388+256*PEEK 16389 i pritisnuti uobičajeno NEWLINE. Ukoliko naš ZX81 nema proširanu memoriju, na ekranu će se ukazati broj 17408 — to je adresa poslednjeg postojećeg bajta u memoriji. Na prvi pogled izgleda da nema nikakve potrebe za ovom sistemskom promenljivom obzirom da računar uvek ima konstantnu memoriju, ali nije tako. Kada uključimo ZX81, on najpre proverava da li je priključen modul od 16 (ili možda 32 ili čak 64) kilobajta i rezultat te provere ubeležava u sistamsku promenljivu RAMTOP. Kada docnije zahtevamo od računara da smesti neki program u memoriju, on neće morati da proverava



Kod skromnijih modela ljubitelji računara najviše tragaju za skrivenim mogućnostima ZX 81

da li je ona dovoljno velika za njegovo smeštanje — jednostavno će pročitati sadržaj RAMTOP-a.

Nema nikakve prepreke da pomoću POKE promenimo sadržaj RAMTOP-a — otkucajmo POKE 16388, 255 NEWLINE POKE 16389, 255 NEWLINE i računar će imati utisak da je priključeno 56 kilobajta memorije. Znači li to da ne moramo da trošimo novac na dodatne memorijske module? Naravno, ne znači. Mi jesmo naveli računar da poveruje da ima više memorije ali time nismo stvorili fizičke jedinice koje je podržavaju — pokušaj da smestimo nešto u nju može da izazove krah našeg računara. Zbog svega toga nema mnogo smisla „podizati" RAMTOP ali njegovo spuštanje može da se pokaže vrlo korisnim.

Spuštanjem RAMTOP-a smo stvorili utisak da radimo sa manje memorije. Zar je to dobro? Odgovor je, začudo, potvrdan: ponekad može da bude korisno da se odraknemo dela memorije. Najbanalniji primer je simuliranje osnovne verzije ZX81. Ukoliko, na primer, želimo da isprobamo neki program koji radi na ZX81 sa 1 K (ili da se uverimo da će neki program koji smo pisali moći da upotrebe i oni koji nemaju proširenje memorije) ne moramo da isključujemo računar i vadimo modul — dovoljno je da otkucamo POKE 16388, O NEWLINE POKE 16389, 68 NEWLINE NEW NEWLINE (pažnje — poslednje naredba će obrisati čitavu memoriju računara, pa program koji se u njoj nalazi treba prethodno da snimite na kasetu) PRINT USR 1040 NEWLINE. Vaš ZX81 je time pretvoren u računar sa samo 1 K memorije!

Trikovi sa vrhom memorije

Sledeća upotreba sistemske promenljive RAMTOP je nešto manje očigledna ZX81, naime, ima ružnu osobinu da, kada se otkuca komanda SAVE, snima na traku sadržaj čitave memorije. Ukoliko je program koji se nalazi u njoj kratak, snimaćemo gomilu praznih bajtova. Što je još gore, kada pokušamo da učitamo neki program sa kasete, računar insistira na tome da upiše i sadržaje svih promenljivih i nizova. Ako, dakle, imamo podatka koje želimo da prosledimo od jednog programa do drugog, moraćemo manuelno da ih prepišemo (ili, ako ga posedujemo, upotrebimo printer) i ponovo otkucamo kada drugi program bude u memoriji. Zar tu ne postoji neki bolji metod?

Naravno, postoji. Računar, kao što smo rekli, posle SAVE snima na traku čitavu memoriju. Ali, kako on zna kolika je ta „čitava memorija"? Razume se, tako što konsultuje promenljivu RAMTOP! Ako u RAMTOP unesemo neki broj manji od realnog, svi podaci koji se nalaze na kraju memorije će biti zaštićeni od brisanja. Ukoliko, veštom upotrebom PEEK i POKE, u taj prostor prenesemo sadržaj nekog niza ili matrice, učitamo nov program i obavimo „inverznu transofrmaciju", izbećićemo prepisivanje i prekucavanje podataka i tako ispraviti jedan od ozbiljnih propusta konstruktora ZX81.

Kao primer upotrebe RAMTOP-a dajemo program koji vam omogućava da u memoriji držite nekoliko sadržaja ekrana i razmenjujete ih prema potrebi. Data su, zapravo, dva programa: jedan je pisan na bejziku ali nije namenjen praktičnoj upotrebi — previše je spor. Ipak, on će početnicima pruž[iti] šansu da shvate kako radi jedan mašin[ski] program, koji može da bude koristan

program za mnoge igre u kojima je potrebna animacija akrana.

```
 10 FOR I=0 TO 21
 20 FOR J=32 to 63
 30 PRINT CHR$ J,
 40 NEXT J
 50 NEXT I
100 LET D=PEEK 16396+256*PEEK
16397
105 LET R=PEEK 16388+256*PEEK
16389
110 FOR I=0 TO 792
120 POKE R+I,PEEK (D+I)
130 NEXT I
200 PRINT AT O,O;
210 FOR I=0 TO 21
220 FOR J=63 TO 32 STEP-1
230 PRINT CHR$ J
240 NEXT J
250 NEXT I
300 FOR I=0 TO 792
310 POKE D+I,PEEK (R+I)
320 NEXT I
```

Assembly code	Hexadecimal
LD HL,(RAMTOP)	2A 04 40
LD BC,(16514)	ED 4B 82 40
ADD HL,BC	09
EX DE,HL	EB
LD HL,(D-FILE)	2A OC 40
LD BC,793	01 19 03
LDIR	ED BO
RET	C9
LD HL,(RAMTOP)	2A 04 40
LD BC,(16514)	ED 4B 82 40
ADD HL,BC	09
LD DE,(D-FILE)	ED 5B OC 40
LD BC,793	01 19 03
LDIR	ED BO
RET	C9

Pra nego što startujete bejzik program, morate da promanite sadržaj RAMTOP-a tako da se na vrhu memorija nađe dovoljno prcatora za jedan ili više sadržaja akrana. Da, na primer, smsetite dva sadržaja akrana, potrebno vam je najmanje 793*2=1586 bajta. Ukoiiko, posle uključivanja kompjutara, otkucata POKE 16389,121 NEWLINE NEW NEWLINE (ovo NEW je uvek potrabno da bi ZX81 rainicijalizovao atak na novu poziciju; ako na znata šta ja stek, zanemarite ovu napomenu obzirom da sa aspekta korianika nije mnogo bitna). Zatim unesite bejzik program, popunita akran nakim rečime i brojevima i startujte program. Demonstracija je, čini nam se, prilično afaktna.

Mašinski potprogram možda nije lako analizirati, ali početnici ovaj posao mogu da ostave i za „arećnija vremena". I semo unošenja ovog programa u prvu REM liniju (???) predstavlja svojevrsnu zagonetku. No, na budimo nestrpljivi: na nakoj od sledečih strana ćemo upoznati način na koji se mašinski programi unose, pa ćemo iskoristiti i ovaj. Oni koji bolje poznaju mašinsko programiranje će unati program u prvu REM liniju od adrese 16516, smestiti na adrese 16514 i 16515 broj koji odgovara oznaci akrana koji smaštaju ili pozivaju (0,0 za prvi akran, 25,3 za drugi i tako dalje) i smestiti ekran u memoriju se RAND USR 16516, a povratiti ga se RAND USR 16534.

Sve sistemske promenljive

Čitava priča nse je prilično udaljila od područja siatemskih promanljivih koja proučavamo. Čini nam se da će većini čitalaca biti od koristi da se vratimo na početak i izložimo mogućncati upotrabe nakih drugih sistemskih varijabli (reč varijabia se u programerskom žargonu koristi daiako češće od iape domaće reči promenljiva iz nepoznatih razloga). Počećemo od semog početka — promanljiva ERR NR koja se nalazi na lokaciji 16384. Kao što joj i ime govori, u okviru ove promenljiva računar amašte kôd greške koja je nestupila. Ukoliko, na primar, vaš program završite se POKE 16384,29, računar će prijaviti inače napostojeću U arror koja, prema časopisu „Sinclair User", može da označi grašku korianika(User arror). Isti časopis predlaže da pokušate da u ovu sistamsku promanljivu smestite brojave 43, 70,72, 73, 74, 75, 61, 82 i 89. Zašto ih na poslušate?

Sistemska promanljiva PPC se nalazi na iokacijama 16391 i 16392. U okviru nje računar smašta linijski broj bejzik naredbe koja se trenutno izvršava. Ukoliko promenite ovaj broj, na izlažete se pravelikoj opasnosti obzirom da se on koristi samo kada računar završava se radom (tada se, na primar, na akranu pojavi 0/123 što znači da se izvršavanje programa završilo linijom 123 pa se u promanljivoj PPC naiazio isti broj). Dakle, ova sistamska promanljiva može jedino da se iskoristi za ispisivanja neobičnih poruka na kraju programa.

Sistamaka promanljiva DF-CC seopštava korisniku mesto na kome će štampati sladeća naredba PRINT. Kao i obično, sistamska promanljiva se sastoji od dve memorijaka lokacije koja treba ujedinitl u jedan broj primenom naredbe LET A=PEEK 16398+256*PEEK 16399. Posle toga, promenljiva A sadrži adresu koja može prilično korisno da se upotrebi. Otkucajmo, na primer, POKE A, 123 i na odgovarajućem mestu akrana će se pojaviti karakter čiji ja kod 123. Ništa naročito, reći ćete, to može da se postigna i naredbom CHR$. Pa dobro, utkucajmo LET B=PEEK A i promanljiva B će sedržati kod karaktera koji se već nalazi tamo gda će siedeća PRINT naredba ciljati. Taj broj možemo da ispitamo i tako otkrijemo da li je, na primar, svamirski brod u nakoj igri izašao iz dopuštenog dala svemira koji ja ograđen dabelim iinijama.

Sistemske promenljiva VARS (16400—16401) i E-LINE (16404—16405) saopštavaju adrese početka i kraja daia memorije koji ja namenjen smeštanju promanljivih koje se koriste u bejzik programu. Obzirom da pranošenje argumenata iz bejzik programa u mašinski moža da se ostvari na daieko jednoatavniji način, ova promenljive nisu previše često u upotrebi.

Na iokaciji 16418 se krije sistemska promanljiva DF-SZ koju početnici „obožavaju". Ona saopštava računaru koliko ja iinija na kraju akrana razarvisano za naredbe koja korisnik kuca. Tamo se normalno nalazi broj 2 ali, uz određen rizik da dođe do kraha, ova promanljiva moža da dobija vrednost 1, čime se efektivno dobija još jedan red. Sa druge strana, ukoiiko nam nedostaje memorija možemo da damo DF-SZ vrednost veću od 2, čime se korisno upotrabljeni deo akrana smanjuje, uz odgovarajuću uštedu mamorijskog prostora.

Sistemsku promenljivu NXTLIN (16425-16426) korista mahom boiji programeri. Ovde se naiazi apsoiutna adresa početka sledeće BASIC naredbe koja može da bude naobično koriana za startovanja mašinskih programa. Ukoiiko se, na primar, u REM liniji broj 101 nalazi neki mašinski program, on može da bude startovan iz linije 100 primanom naredbe LET A=USR (PEEK 16425+256*PEEK 16426+5) (broj 5 potiče otuda što iinijski broj i druge informacija koja sadrži početak programake linije 101 zauzimaju čitavih pet bajtova).

Završimo ovu kratku priču o sistemskim promenljivima adresama onih koja namaju nikakvu funkciju: 16417, 16507 i 16508. Ukoliko na posedujete printer ili ovaj nije prikijučen, slobodno koristite i aistemske varijable na adresama 16444-16476. Ovako dobijenih 35 bajtova možate da upotrebite za smeštanja podataka za koje nemate gde drugde da rezarvišete prostor.

Gde smestiti mašinac

Upoznali smo aistamska promenljive, pa možemo da se orijentišemo na mašinske programa. Prvi problem je gde da ih smestimo. Obzirom na izloženo, učinilo bi se da ja pravo mesto za ova programa iznad RAMTOP-a — tamo ih neće ometati niti pomarati nikakve izmana bejzik programa, pa čak ni naredbe NEW. No, pcatoje i loše strana lokacija iznad RAMTOP-a — na mogu da budu snimljene na kasetu se matičnim bejzik programom. Ostaja nam, dakle, da ih smestimo iza bejzika? Ni to nije naročito dobro jer bi avaka promena bejzik programa (a promane su česte, pogotovu u toku sastavljanja i testiranja programa) pomerila mašinski program i tako učinila da on prestana da radi (većina JP naredbi, na primer, sedrži apsolutne adrese, što podrazumeva da program neće biti pomaran). Ostaje nam samo da mašinski program smestimo unutar bejzika, nadajući se da temo neće nikoma smatati. Ali, kako onda da ga „sakrijemo" tako da ga bejzik interpratator na primeti, što bi izazvalo prijavljivanja greške? Jednostavno, amestićemo mašinski program u prvu REM naredbu i to pomoću programa koji je poznat kao „Machine Coda Loader" ili program za unošenja mašinskih programa. Upotrabe programa je sasvim jednostavna: otkucaćemo najpre REM naredbu koju prati ber onoliko jednakih slova koliko će mašinski program imati instrukcija, a zatim RUN. Unosićemo mašinski program bajt po bajt, koristeći heksadekadni sistam (heksadakadni sistem je nešto što na bi trebalo da predstavija napoznanicu za ona koji čitaju ovo poglavije). Po unošenju mašinskog programa možemo i da ga startujemo otkucavši RAND USR 16514.

```
 1 REM MMMMMMMMMMMMMMM...
10 LET T=16514
20 CLS
30 INPUT H$
40 POKE T, (CODE H$(1) — 28)*
   16+CODE H$(2) — 28
50 T=T+1
60 GOTO 20
```

Svaki čitalec ovih redove je verovatno poželeo da isproba novi program za unošenja mašinskih programa. Zašto da ne? Sledeći program koji objevljujemo će u mnogoma unapredit tastaturu vašeg računare uvodeći funkciju poznetu pod imanom autorepeat! To, praktično, znači da će duži pritisak ne neki taster naterati računar da ne ekranu proizveda efekat kao da je taster bio pritisnut više puta što, na primer, može da bude korisno za iako krairanje REM naredbi za unošenja mašinskih programa koja ćete semi sasteviti. Kade unesete sledači program, otkucajte uobičajeno RAND USR 16514 i u gornjem levom uglu ekrana će se pojaviti kvadrat koji će treperiti sve dok je autorepeat aktiven. Pritisnite, na primar, taster A i posmetrajte akran. ZX81, izgleda, ume svašta, treba mu to samo dobro objasniti.

```
4082 DD 22 7B 40 DD 21 8B 40 C9 E1 D1 C1 F1 22 3E 40
4092: 21 AB 40 E5 2A 3E 40 F5 C5 D5 E5 B6 D3 FD 2A OC
40A2: 40 CB FC DD 2A 7B 40 DD E9 DD 22 7B 40 DD 21 8B
40B2: 40 E5 F5 3A 34 40 CB 5F 28 08 2A OC 40 23 7E C6
40C2: 80 77 2A 25 40 24 2C 7C B5 20 07 3E 32 32 21 40
40D2: 18 15 24 7C B5 28 F4 3A 21 40 A7 28 06 3D 32 21
40E2: 40 18 04 AF 32 27 40 F1 E1 C9
```

Zaštita i kako je skinuti

Sledeći redovi će biti posvećeni radu sa kasetofonom ili, praciznija rečeno, zaštićenim programima. Skupiji kompjuteri najčešće imaju manja ili više efikasne matoda zaštite programa od neovlašćenog prepisivanja ili, kako se to u žargonu kaže, „piratovanja". ZX81, doduše, neme posebne naredbe kojima se ove zaštite ostvaruje ali od pomoći može de buda jednostavno smeštanje naredbe SAVE u program. Vratimo se, ne primer, ne prethodni program ze automatsko ponavijanja i zamislimo da želimo da ga damo nekome ko neće smeti deije da ge presnimave. Unećemo ovej program u prvu REM liniju, dodati program
2 SAVE „REPEAT"
3 RAND USR 16514 i otkucati RUN. Računer će zanemariti REM liniju, neići na SAVE i snimiti čitav program na kasetu. Kade onaj kome ste deli program unese kasetu i otkuce LOAD"", računar će učitati program i primatiti po nekim informacijema koje su zapisane uz program da se SAVE koja ga je snimilo nalazilo u programu. ZX81 je programiren tako de automatski nastevi izvršavanja ovekvog programa i, po nailasku ne naredbu broj 3, „skoči" na mašinski potprogram koji se nelazi u REM iiniji. U našem primeru mašinski program će trenutak docnije prestati se radom pa će njegov vlasnik bez probiame moći da ge ponovo snimi na isti način. No, šta da se u mašinskom progremu nija nelazila ni jedne HLT odnosno RET naredba? On bi se neprekidno izvršavao, eli njegov korisnik ne bi mogao de ga zaustavi da bi ga presnimio. Lukavo, zar ne?

Svima nam ja pozneto da kod nas, i pored ovekve zaštita, ima dosta preprodavaca koji na neki „tejanstven" nećin uspevaju da prakopiraju i preprodaju programa. Kako im to uspeva? Neki poseduju mašinski program koji imitira neredbu LOAD i unosi program u meoriju bez njegovog stertovenje. No, iskusnijim programerima ja poznat dalako boiji trik: prebecimo računar u FAST mod i otkucajmo RAND USR (836). Očekujate da su uputstva tek počela? Greška — uprevo su završena. Računer je „praveren" tako da učita mašinski i bejzik program i ne startuje ih! Jedno SAVE ća ga zatim snimiti ne neku drugu traku.

Simulacija funkcije VERIFY

Još jedan trik vezan za snimanja na kasetu je simulacija funkcije VERIFY. Ova funkcija, kao što je vlasnicima Sectruma, BBC, TRS 80 i drugih kompjutera poznato, služi de se proveri de li ja upravo snimijani program korektno zapisan. ZX81, na žalost, nija opremijen ovom funkcijom, što znači da poatoji određena mogućnost da program koji sta čitavog dana radili biti nekorektno snimljen i tako zanavek izgubljen. Moguće je sasteviti mašinski program koji bi verificirao ono što je snimijeno (nevoija ja samo u tome što bi i sam progrem za verifikaciju morao da se snima zajedno sa svakim drugim programom, jer ZX81 snima čitavu memoriju) ali postoji i bolji (ne žalost i menja siguran) trik. Kada snimite program, pramotajte kasetu na njegov početak i otkucajte LOAD „ime programa", pa pritisnite NEWLINE i stertujte kasetofon. Ako se posle izvesnog vremene pojavi poruka 0/0, postoji značajne mogućnost de je računar korektno učitao i snimio program. Na taj nečin je provereno da li ja ime programe koraktno prepoznato, a za slučaj da nije, nikakav program nije učiten, pe možete da pokušate de ponovo animite vaš program (on se i deije nalazi u mamoriji) ne neku drugu kasetu. Nedostatak matoda je u tome što postoji mogućnost da ja traka bile fizički oštećena i da je, premde je početak korektno upisen, nastupila greška u sredini programa. Dovoljno je da jedan bajt ne odgovara istini i program će verovetno propasti!

spectrum i njegove neobičnosti

Iako verovatno omalovažavaju prethodnika ovog računara, vlasnicima Spectrume bismo ipak preporučili da najpre prouče prethodno poglavlje o ZX81. Razloge nije teško pogoditi: ova dve računara su, i pored toga što je Spectrum znatno moćniji, relativno slična i činilo nam se nepotrebnim de ponovo počinjemo od samog početka, koji se odnosi na organizaciju memorije Spectruma i takozvane sistemske promenljive. U ovom tekstu ćemo se baviti uglavnom njegovim neobičnostima.

Memorije Spectrume ja organizovane nešto komplikovanije od memorije ZX81 pošto u njoj more da se neđe više tipove podataka a i slika u boji. To je uslovilo uvođenja novih slatemskih promanljivih koje su sažete u prvoj tabeli. Pogied na nju otkriva da ROM Spectrume ime 16 kilobajte i zauzima adrese od 0 do 16383. Vlasnici Spectruma opremijenog osnovnom memorijom od 16 K mogu da raspolažu RAM-om koji se pruža između adresa 16384 i 32767, a oni koji su kupili proširanu verziju — do 65536. Sav ovaj prostor, ipek, nije sasvim sloboden — za normalan red Spectruma potreban je određeni RAM koji se automatski rezerviše svaki put kada uključite računar. Najveća ovekve stevka je ekran koji oduzime 8 kilobajte memorije, o čijoj ćemo organizaciji govoriti docnije. Počevši od adrase 23296 smešten ja befer koji koristi štampeč, kada on nije priključen, ovaj prostor ostaje neiskorišćen. Iza njega sledi 256 bajtove koji sadrže sistemske promanljive i određen broj bajtova koje interno koristi mikrodrajv.

Trikovi sa ekranom

Sve dosadašnje stavke u tabeli su imala fiksne adrese, a sada stupaju na scenu sistemska promenljive. Kraj područje posvećenog mikrodrejvu nije fiksan, već je smešten u sistemsku promenljivu koja se zove CHANS i smeštena ja u marmoriji na adresama 23531 i 23532. Iza toga se nalaze bejzik programi, korišćene promenijive i nizovi, iinija koje se trenutno kuca ili edituje, mašinski stek i, na kreju, RAMTOP. Kod Spectruma RAMTOP ne označava kraj memorije već kraj područja u koja se smeštaju podeci koja koristi neki bejzik program. Iza RAMTOP-a se nalazi područje u koje računar smešta podatke od značaja za generisanja posebnih karaktera koje je korisnik ranije definisao.

Najteža shvatljive oblast organizacije Spectrumove memorije ja ekran. Kad što je poznato, akran se sastoji od 24 iinije od

kojih svaka sadrži po 32 karaktera Svaki karakter je predstavljen određenim brojem tačaka, u normalnom stanju, to je matrica 8 8 Primenom naredbe PLOT, korisnik može da osvetli svaku od tako dobijanih 32 8 24 8=49152 tačke, dajući joj jednu od osam boje Da li je poslednja rečenica tačna? Jednostavan račun uveriće nas da nije!

Da kontrolišemo boju svaka tačke treba da utrošimo 3 bite (tako da, na primer, 000 označava plavu boju, 001 zelenu, 010 crvenu i tako dalje), što znači da nam je za ekran od 49152 tačka potrebno 147456 bita ili osamnaest kilobajta Pošto se za ekran koristi samo 6,75 kilobajte, jasno je da je primenjen neki trik radi uštede memorije ,Trik'' je izveden na sledeći način: u prvih 6 kilobajte RAM-a svakoj tački je dodeljen po jedan bit koji govori računaru da li je ta tačka osvetljene bojom koja je označena kao INK ili bojom koja je označena kao PAPER (1 odgovara boji ,,mastila'', a 0 boji ,papira'') Pa dobro, reći ćete, kako računar zna kojom je bojom ,,ofarban'' svaki karakter i, što izgleda još važnije, da li on treperi ili ne? Jednostavno, 768 bajtova memorije koji sa nalaze na adresama 22528 — 23296 imaju funkciju ,,atribute'' Svaki od ovih bajtova odgovara jednom karakteru (32 24=768) i dobija vrednost u skladu sa sledećom formulom:

atribut =128 F+64 B+8 P+1

gde je sa F označen broj 0 ili 1 koji odgovara mirovanju odnosno treperanju nekog karaktera, sa B 0 ili 1 u zavisnosti od toga da li je neki karakter uopšte osvetljen ili nije, sa P broj između 0 i 7 koji označava boju ,,papira'' (prema redosledu na tastaturi, 2 je, na primer, crvena boja) a sa I broj između 0 I 7 koji ogovara boji ,,mastila''

Bitovi u haosu

Površan pogled na rezonovanje koja smo izložili uliva neverovatno poštovanje prema konstruktorima Spectruma uštedeno je nekih jedanaest kilobajta boljom organizacijom memorije! No, kratko razmišljanje nas uverava da i u programiranju važi ,,zakon održanja anergije'' (ovde zakon održanja memorije) da bi se memorija organizovala na racionalniji način, morali su da se učine i neki ustupci Ustupci se sastoje u tome da ne možemo slobodno da raspolažemo bojom svake tačka, već sve tačke koje čine jedan karakter moraju da budu isto obojene Da li bi bilo bolje da je utrošeno još 11 kilobajta i postignuta kontrola svaka tačka ostaje pitanje za diskusiju — Sinclair se nije odlučio za ovakvo rešenje s obzirom da bi tako morao da zaboravi na ZX Spectrum sa 16 K RAM-a

```
100 FOR I=0 TO 703
110 PRINT ,,■'';
120 NEXT I
130 SAVE ,,Slika'' SCREEN$
140 CLS
150 INPUT ,,Premotaj kasetu,
    pritisni PLAY i NEWLINE'';Z$
160 LOAD ,,Slika'' SCREEN$
```



Tehnička elegancija i paradoksi ZX Spectrum

Mapa memorije

Početne adresa ili ime sistemske prom.	Položaj sistemske promenljive	Sadržaj memorije
16384	—	Ekran
22528	—	Atributi
23296	—	Bafer za štampač
23552	—	Sistemske promen.
23734	—	Mikrodrajv
CHANS	23631	Kanali
PROG	23635	BASIC program
VARS	23627	Promenljiva
E_LINE	23641	Tekuća linija
WORKSP	23649	Bafer za INPUT
STKBOT	23651	Stek za računanje
STKEND	23653	Slobodan prostor
sp+)	—	Mašinski i GOSUB stek
RAMTOP	23730	Kraj BASIC područja
UDG	23675	Generisani karakteri
P_AMT	23732	Kraj RAM-a

+) ap označava registar procesora Z80 a ne sistemsku promenljivu

```
100 INPUT ,,Početne adresa'';T
110 INPUT Z$
120 IF Z$=,,S'' THEN STOP
130 PRINT Z$, '' '';
140 LET Z$(1)=CHR$ (CODE Z$ (1)-7* (CODE Z$(1)>57))
150 LET Z$ (2)=CHR$ (CODE Z$ (2)-7* (CODE Z$ (2)>57))
160 POKE T, 16*CODE Z$(1)+CODE Z$(2)-816
170 LET T=T+1
180 GOTO 110
```

Voleli bismo da ovim možemo da završimo izlaganja o akranu ali ne smemo da pranebegnemo ,,sitnicu'' koju nije lako objasniti već smo rakli da je svakoj od 256 192=49152 dodeljen jedan bit koji reprezentuje njeno stanje Tako dobijeni bitovi zauzimaju prvih 6 K memorije ali organizovani ne, kratko rečeno, neshvatljiv način. Sve će postati jasno (ili nejasno) ako otkucate i izvršite sledeći bejzik program

Računar najpre ispunjava akran punim karakterima i snima ga na kasetu Zatim ekran biva izbrisan i počinja unošenja njegovog sadržaja Ovo unošenje traje toliko dugo da ćete primetiti da se akran ne popunjava odozgo nadole, sleva nadesno ili na bilo koji način koji možete da očekujete, nego po sekcijama koje sa sastoje od po osam redova karaktera Karakteri sa, opet, ne unose celi nego liniju po liniju računar najpre ispisuje prvi red tačaka prvog reda karaktera, zatim prvi red tačeka drugog reda i tako dalje sve dok na stigne do osmog reda Zatim se vraća nazad i ispisuje drugi red tačaka koja čine prvi red karaktera, prelazi na drugi red i tako dalje sve dok ne završi prvu sekciju ekrana kada prelazi na drugu i, na poslatku, na treću Bitovi koji odgovaraju karakterima su raspoređani na isti ovakav nevarovatni način

Ako vas je od prathodnog objašnjenja zabolela glava, na brinite — ni vrhunski programeri ne mogu lako da ga ,,svara'' Napisati mašinski program koji će osvetliti neku određanu tečku na akranu nije lako, pa se u već većini komercijalnih programa

pozivaju gotove rutina iz ROM-a — nećemo trošiti vreme na njihov opis obzirom da pretpostavijamo da posedujete knjigu Spectrum ROM Disassembled''

Četrdeset karaktera

Pravi ja trenutak da boljim programarima uputimo i jedan mali izazov. možate li da sastavite program koji će da omogući stvaranja 40 karaktera u svakom redu? Program namerno nećemo dati, ali vam mogu koristiti naki praktični saveti

Pretpostavimo da žalite da štampate sadržaj alfanumaričke promenljive a$ koja sadrži 40 znakova u jednoj jedinoj liniji ekrana Treba, pre svaga, da pronađete promenljivu a$ u području koja ja određeno za čuvanja bejzik varijabli To ćete najlakše uraditi konsultujući ponovo knjigu „Spectrum ROM Disassembled'' i rutinu LOOKVARS koja počinja od adrase 10418 Kada pronađete adresu prvog bajta promenljiva a$, nađite CODE tog karaktara, a zatim i osam bajta podataka koji se nalaze u ROM-u i određuju oblik tog slova (tabela se oblicima slova u ROM-u započinja od 15616). Formula koju pri tom koristite ja „oblik karaktere počinja od edrese 15360+8*CODE (a$(1))''.

Pošto locirate potrebnih osam bjatova, smestite ih redom u prvi, trideset treći, šezdeset peti, devedeset sedmi itd bajt područja mamorije koja je označeno keo „bafer za štampeč'' Ovo područje može zgodno da se upotrabi obzirom da sadrži baš onoliko bajtova koliko i jedna linija od 40 karektera

Zatim nađite CODE drugog karaktera, odgovarajućih osam bajtova koji govore o njegovom obliku i pranesite ih u drugi, trideset četvrti i tako dalje kerakter bafara za štampač Kada to učinite, pomerite svaki bajt ulevo za po dva bite i pređite na treći karaktar niza a$ Kada njegov oblik pranesate na odgovarajuća meste, pomarite svaki bajt bafera ulevo za četiri bita pa pređite na sledeći karakter Kada isto učinite sa svih 40 karaktere, problemi su se približili kraju Ostalo ja „samo'' da prenesete sadržaj bafara u deo mamorije koji ja namenjan akranu i to na na prirodan način, vodeći računa o, blago rečeno, čudnom rasporedu za koji se konstruktor opredalio

Na akranu mogu da nađu mesto pedesetak karaktera (da budemo precizni, 51 obzirom da sa svako slovo relativno pristojno predstavija matricom 5 7 i da ja 51 5=255 tj skoro 256) pa i više, do 64, ali tada neka slova gube prirodni oblik Da bismo izveli bilo koju od pomanutih operacija, moramo da upoznamo mašinski jazik Spectruma Nemamo namaru da sa ovom prilikom praviše upuštamo u njagovo objašnjavanja (ako za to ima interesa, možemo da pokrenamo Školu mašinskog jezika u „Galaksiji''), ali ćemo svakeko objasniti način na koji mogu da se unose mašinske instrukcije i dati nekoliko primara.

Mašinac u REM liniji

Kod ZX81 mašinski programi sa, kao što znamo, smeštaju u prve REM naredbe programa. I kod Spectruma može da se pribegna istom rešenju koja ima nakoliko odličnih strana (najbolje ja ta što se bejzik i mašinski program istovremeno učitavaju u mamoriju), ali koja je prilično nepopularno među vlasnicima Spectruma iz razloga koji autoru ovog teksta nisu naročito jasni Verovatno je da ja Spectrumova mogućnost da, pomoću naredbe SAVE . . CODE snima bilo koji sagmant mamorije na traku podstakla programare da mašinski program smeštaju iza bejzike i da njegovo učitavanja povare bejzik programu koji je na traku smešten ispred njaga Bilo keko bilo, mašinski program možate da unesete pomoću programa koji je relativno sličan odgovarajućoj rutini za ZX81, jedina principijelna razlika ja proistekla iz činjanice da Spectrum koristi ASCII, a ZX81 naki specijalni kod za smeštanja karaktera. Ukoliko je, kao što ja to često slučaj, mašinski program dat kao decimalni (a na heksadakadni) damp, možemo da zamenimo naredbe 20, 25, 30 i 35 jednostavnim POKE — tada prevođenja haksadekadnih vrednosti nije potrebno i moža samo da izazove probiama

Iskoristimo novi program i unesimo mašinsku rutinu koja ća ga testirati Kada smo već kod testiranja, na bi bilo loše da program koji ćemo sastaviti testira mamoriju našeg spectruma i uveri nas da krivicu za sva naše programa koji na reda na snosi „naispravni računar'' Test ja vrio jednostavan najpre ćemo u svaki bejt memorije upisati broj 00 i proveriti da li je svaki njegov bit „resetovan'' a zatim ćemo upisati FF i provariti da li je on „setovan''. Ne možemo, jasno, da upisujamo podatke u **svaki** bajt RAM-a obzirom da bi računar krahirao kada bismo porematili sistemske promenljiva, ali možemo da provarimo „slobodan'' deo RAM-a — da ja mamorija naispravna kod nake od sistemskih promanljivih, računar verovatno na bi ni radio

Program dat u sledećoj tabeli smestićemo, naravno, u bafar za printer (počevši od adrese 23296) i startovati sa PRINT USR 23296 Ukoliko ja sva u redu, nakoliko trenuteka docnije će se na akrenu pojaviti broj 32575 (65343 za mašina sa 48 K) koji indicira da ja računar ispravan Ukoliko nije, ponovite čitav test proveravajući da li ja program dobro unesen. Ako jeste, vaš računar ja naispravan ili (što je nešto varovatnije) u našem specijalnom izdanju postoji bar **jedna** štampareka greška.

Hax coda	Assembiar coda
2A 65 5C	LD HL, (23653)
44	LD B,H
4D	LD C,L
3E OO	LD A,O
7E	LD (HL), A
77	LD A, (HL)
FE OO	CP O
CO	RET NZ
3E FF	LD A,255
77	LD (HL),A
7E	LD A,(HL)
FE FF	CP 255
CO	RET NZ
A7	AND A
23	INC HL
ED 72	SBC HL,SP
C8	RET Z
A7	AND A
ED 7A	ADC HL,SP
O3	INC BC
18 E7	JR −25

Ekskluzivni RESET

Pisanja mašinskih programa ja, kao što ćeta se ubrzo uvariti, „hlab sa sedam kora'' Ako u bejzik programu pogrešite, neće se desiti ništa loše — pritisak na BREAK će na akranu dovesti poznatu poruku READY i moći ćete da ispravite grešku i ponovo startujete program Ako se, međutim, u vašem mašinskom programu nađe instrukcija tipa

CIKLUS JR CIKLUS

naći ćeta se u velikoj nevolji. Čak i onoma ko na poznaja mašinski jezik nije teško da zaključi da se računar, izvršavajući ovu inatrukciju, staino vrti u „proceduri'' koju na možemo da prekinemo pritiskom na BREAK — ovaj taster je pri izvršavanju mašinskog programa potpuno „iskijučen''. Jedini izlaz je da iskijučimo računar iz mraže i trenutak docnije ga ponovo uključimo Tada će sva biti u redu, ali će i svi programi iz memorije biti napovratno izgubljeni

Vlasnici Spektruma su na razna načina izbegavali ovaj problam naki su svaku varziju mašinskog programa snimali na kasetu kako gubitak mamorija na bi bio tragičan, ali ovako rešenje izaziva priličan gubitak vremena Drugi su sa dosetili da u mašinskim programima vrlo često testiraju da li je taster BREAK pritisnut u nadi da će „krah'' nastupiti zato što računar naprakidno izvršava istu veću grupu instrukcija, treći su koristili **RESET** nožicu procesora Z80 da bi, preko jednog tastera, obezbedili hardvarski prekid rada, u čemu su imali manja ili više uspeha, a četvrti su izabrali najlakše rešenja nisu ni piseli mašinske programe Na žalost, navolja nastaja i kod bejzik programa koji, na primer, izgleda ovako

10 INPUT LINE A$ GO TO 10

Startujte ovaj progrem i pokušajte da ga prekinate To će biti veoma teško — skoro namoguće

Zahvaljujući Englazu Toniju Bejkeru (Toni Baker) svi probiami ovoga tipa mogu da postanu neprijatna prošlost U prvom broju časopisa Your Spectrum (januar 1984, ovo ja, inače, akskluzivna informacija koja ja ušla u naše specijalno izdanje u poslednjam trenutku, pola sata pre konačnog zaključivanja) objavljen je, naime, program koji bez ikakvih hardvarskih intervencija omogućava prekid izvršavanja bilo kog mašinskog ili bejzik programa jadnostavnim pritiskom tastera!

Program koji dajemo treba da smestite u memoriju vašeg Spectruma (mogućnost, na žalost, postoji samo za ona koji poseduju Spectrum 48 Kb) počevši od adrese 80E2 koristeći hex-loadar Program ima samo 89 bejta, pa njagovo unošenje na pradstavija baš nikakav problam probajte ga čak i ako niste počeli da pišete mašinske programa Novi **RESET** će biti aktiviran ako otkucate **RANDOMIZE USR 33001** Kada vam god zatreba da izvršite **RESET**, jednostavno istovremano pritisnite tastere **CAPS SHIFT** i **ENTER** Na akranu ća se pojaviti odgovarajuće poruke i računar će biti spreman za normalan rad! Ukoliko želite da iskijučite **RESET**, otkucajte **RANDOMIZEUSR32994** Čini nam se da nećeta osetiti potrebu da bilo kada iskijučujete ovu novu proceduru i da ćeta se ućitavati u memoriju svaki put kada uključite vašeg Spektruma

Nova naredba **RESET** funkcioniše čak i u toku izvršavanja nakih internih funkcija računara. Otkucajte, na primer, neki bejzik

program i pritisnite ENTER da bi ovaj bio izlistan U toku listanja pritisnite RESET i ono će trenutno biti prakinuto Naredba RESET ima i par minornih ograničenja

- Izvršavanja naredbe NEW će deaktivirati RESET čak i ako je ovaj program smešten iznad RAMTOP-a Naredba RANDOMIZE USR 33001 će ga, jasno, ponovo aktivirati ako nija izbrisan
- RESET ne radi dok računar izvršava SAVE, LOAD ili BEEP Tada je, naime, interapt isključen i trik na kome je program zasnovan više ne funkcioniše. Ukoliko se u vašem mašinskom programu na samom početku nalazi instrukcija DI (ne zaboravite na EI pre povratka u bejzik), RESET neće moći da ga zaustavi. Zašto biste želeli da to onemogućite? Jedino zbog eventualne zaštite programa, koja je sledeća na našem dnevnom redu

Mašinski kod Asembler

```
                                  DRG 8DE2
3E3F       DEACTIVATE   LD A,3F
ED47                    LD I,A
ED5B                    IM 1
C9                      RET
3E80       ACTIVATE     LD A 80
ED47                    LD I,A
ED5E                    IM 2

C9                      RET
524553     RES_ MESSAGE
45542085
78656375
746564A1                DEFM RESET executed!
0181       I   ADDR     DEFW TEST__RESET

F5         TEST  RESET  PUSH AF
3EFE                    LD A FE
DBFE                    IN A (FE)
1F                      RRA
3807                    JR C NO  RESET
3EBF                    LD A BF
DBFE                    IN A (FE)
1F                      RRA
3004                    JR NC RESET
F1         NO  RESET    POP AF
FF                      RST 38

ED4D                    RETI
2AB25C     RESET        LD HL (RAMTOP)
2B                      DEC HL
F9                      LD SP HL
2B                      DEC HL
2B                      DEC HL
223D5C                  LD (ERR  SP) HL

AF                      XOR A
32715C                  LD (FLAGX),A
CD0116                  CALL CHAN  OPEN
CD6E0D                  CALL CLS  LOWER
213B5C                  LD HL FLAGS
CB9E                    RES 3 (HL)
23                      INC HL
CBEE                    SET 5 (HL)

AF                      XOR A
11EF80                  LD DE RES  MESSAGE 1
CD0ADC                  CALL PO  MSG
FB                      EI
C3A912                  JP MAIN  1
```

kontrolor snimanja

Mehanizam zaštite komercijalnih programa je znatno komplikovaniji i, samim tim, teži za „probijanje" od metoda koji se primenjuju na ZX81. Ipak, dokazano je da se program pisan za Spectruma ***ne može****, zaštititi od presnimavanja na način koji nije savladiv Zaštićeni programi su prava zagonetka za one koji uče programiranja i nepotrebno mistifikuju računar. Ipak, čini nam se da nema velikog smisla trošiti vreme i imaginaciju za unapređenja zaštite koja pred iskusnim korisnikom pada u vodu. Ovo „iskusni korisnik" može da se odnosi i na vas — dovoljno je da pročitate pasuse koji slede. Objavljivanjem programa za skidanje zaštite nemamo cilj da vam damo u ruke „kalauz" za komercijalne programe već korisnu alatku za brzo sređivanje svoje postojeće biblioteke programa.*

Mehanizam zaštite se zasniva na automatskom startovanju programa. Ukoliko se mašinski program nalazi u okviru bejzika (na primer, u nekoj REM liniji), skidanje zaštite je trivijalno: otkucamo MERGE"". STOP i startujemo kasetofon. Po učitavanju programa, njegovo izvršavanja će biti prakinuto (ili, da budamo precizniji, neće ni započeti), pa ćemo moći da ga snimimo na kasetu primenom obične naredbe SAVE

Pravi problem, međutim, predstavljaju mašinski programi koji su snimljeni sa SAVE ... CODE i koji se automatski startuju Da bi se njihovo presnimavanja uspešno obavilo, moramo u memoriju da unesemo poseban program poznat pod imenom „brejkar" ili, kulturnije, „save controler" (kontrolor snimanja) Jedan takav program je sastavio Bogdan Kosanović iz Beograda koji nam je pružio i značajnu pomoć pri realizaciji ovog poglavlja. Program ovde objavljujemo uz neophodna uputstva za njegovo kucanje, snimanje i upotrebu.

Kucanje programa

Najpre otkucajte bejzik program koji je prvi naveden i snimite ga na kasetu sa SAVE „Breaker 3 2" LINE 10 Program je snimljen tako da se automatski startuje, razlog za ovakvo snimanja nije u toma što sam program treba da bude zaštićen nego u toma što je njegova upotreba znatno jednostavnija kada ne moramo da kucamo RUN posle svakog učitavanja

Po uspešnom snimanju i verifikaciji programa izvadite kasetu, ubacite neku drugu (ovde će te kaseta biti upotrebljena samo kao pomoćna), otkucajte NEW i unesite bejzik program dat na sledećoj slici. Ovaj program ne trebe da snimate — on je namenjen jedino unošenju mašinskog programa koji će predstavljati integralni deo „breakara". Pošto dobro proverite ispravnost DATA linije, startujte program sa RUN Trenutak docnije računar će zatražiti da startujete kasetofon i tako snimite mašinski program

Ostala je još jedna stepenica Otkucajte NEW i unesite sledeći kratki BASIC program i startujte ga sa RUN Moraćete da premotate pomoćnu kasetu i u memoriju unesete program koji je upravo snimljen Posle toga možete da odstranite pomoćnu kasetu i, po zahtevu računara, snimite finalni mašinski program iza bejzika koji se već snimljen na prvu kastetu Neprijatni deo posla je završen i možete da počnete da koristite brajker

Napominjemo da su ovolike manipulacije bile apsolutno neophodne. Osnovna karakteristika koju dobar „save controlar" treba da ima je da bude što kraći da bi pomoću njega mogli da se presnimavaju što duži programi. Nije, na žalost, moguće odmah smestiti mašinski program na pravo mesto pošto bi se kosio sa bejzik programom koji ga unosi Zato on najpre biva snimljen na više adresa da bi, pomoću drugog programa, bio „spušten" na pravo mesto Autoru programa se činilo da njegovo unošenje na ovaj način oduzima vreme samo jednom i da se docnija program koristi uz veće mogućnosti

Presnimavanje . . .

Kada zaželite da presnimite neki zaštićeni program, unesite breaker sa LOAD Pošto se program uspešno učita, na ekranu će se pojaviti „meni", odnosno spisak opcija koje možete da upotrebljavate

Ako ste početnik namenjene su vam opcije **bp**, **sp** i **vp**. Da biste uneli neki program u memoriju otkucajte **bp** (uz uobičajeno ENTER) i startujte kasetofon Po učitavanju programa, računar će od vas zatražiti da umetnete kasetu na koju program trebe da se snimi, završite snimanje a onda se od vas očekuje da premotate kasetu da bi se izvršila verifikacija snimka Posle toga možete ponovo da otkucate **bp** i tako unesete neki drugi program ili da otkucate **sp** i tako snimite program koji se i dalje nalazi u memoriji na neku drugu kasetu Posle ovoga možete da upotrebite **vp** i tako verifikujete novi snimak. Opcije **sp** i **vp** možete da ponavljate koliko je god pute potrebno

Analiza programa

Sledeći stupanj poznavanja Spectruma vam dopušta da koristite opciju **ap.** Po njenom izvršavanju računar unosi program sa trake i smešta ga u memoriju od adrese 25500 (dekadno) Posle toga možete da prekinete rad „brejkara" (sa CAPS SHIFT 6). i analizirate uneseni mašinski program pomoću disasemblera ili nekog sličnog programa Treba da vodite računa o jednoj činjenici: program je veštački učitan tako da počinje od adrese 25500 što ne znači da on treba tamo da se nalazi da bi se korektno izvršavao Štaviše, ako startujete mašinski program sa RANDOMIZE USR 25500, skoro je sigurno da će uslediti blokiranja računara i da ćete morati da ga isključujete i ponovo uključujete Opcija **lp** je, dakle, namenjena samo analizi programa, dok za njegovo učitavanja „na pravo mesto" morate najpre da upoznate struktu-

```
10 POKE 23624, VAL "13": PAPER PI/PI:
INK VAL "5": CLS
20 CLEAR 25000: LOAD "" CODE  LET q=
VAL "30"
30 DIM A$( VAL "2"): CLS: PRINT "bp:B
reak program" ' ' "lp: Load program" ' "v
p.Verify program" ' "sp.Save program"
' "lh-Load header" ' "vh:Varify header"
' "ah; Save header" ' ' "lb:Load block" '
"vb:Verify block" ' "sb:Save block" ' '
"th:Show header"
40 INPUT LINE A$: LET H=PI/PI: LET L=
H: LET B$=A$(H): LET C$=A$(H+H): IF C$="
B" OR C$="b" THEN LET H=NOT H
45 IF C$="P" OR C$="p" THEN LET H=-H
50 IF B$=,,S" OR B$=,,s". THEN LET L=NOT
L
60 IF B$="V" OR B$="v" THEN LET L=L+L
70 IF B$="T" OR B$="t" THEN LET L=-L
75 IF B$="B" OR B$="b" THEN LET L=INT
PI+L
80 GO TO CODE ,,n"+VAL ,,20"*L
90 CLS FOR I=25016 TO 25034: PRINT I
, PEEK I;"    ";: IF PEEK I>CODE " " AND PE
EK I<CODE "RND" THEN PRINT CHR$ PEEK I :
GO TO CODE "d"
98 PRINT " "
100 NEXT I : INPUT "Change (Y/N) "; LIN
E B$  IF B$=,,Y" OR B$=,,y" THEN GO TO COD
E "SCREEN$"
102 GO TO q
110 IF H=H-H THEN PRINT USR 25204: GO
TO q
120 IF H=PI/PI THEN RANDOMIZE USR 2516
2: GO TO q
125 PRINT USR 25116: GO TO q
130 IF H=H-H THEN RANDOMIZE USR 25179:
PRINT USR 25060: GO TO q
135 IF H=PI/PI THEN RANDOMIZE USR 2503
5: RANDOMIZE USR 25150: GO TO q
140 PRINT USR 25096: GO TO q
150 IF H=PI/PI THEN PRINT USR 25050: G
O TO q
160 IF NOT H THEN PRINT USR 25071: GO
TO q
186 PRINT USR 25124: GO TO q
170 INPUT "Address= ";L ' "Value= ";H:
POKE L,H: RANDOMIZE USR 25104: GO TO COD
E "Z"
190 PRINT USR 25088: GO TO q
```

```
10 DATA 221, 33, 169, 97, 17, 17, 0, 175, 55,
205,86,5,48,242,201
20 DATA 221,33,166,97,17,17,0,175,24,
22,221,42,182,97,237
30 DATA 91,180,97,55,24,9,221,42,182,
97,237,91,180,97,167
40 DATA 62,255,205,2,8,198,137,98,205
,8,98,205,26,98,24
50 DATA 28,205,203,97,205,88,98,24,
212,17,185,97,33,202,97
60 DATA 1,17, 0, 237, 184, 201, 205, 16, 98,
205,74,98,24,83,205
70 DATA 216,97,24,198,62,253,205,1,22
,175,17,161,9,205,10
80 DATA 12,33,60,92,203,238,205,212,
21,201,33,169,97,17,186
90 DATA 97,1,17,0,237,176,201,205,41,
98,221,33,169,97,17
100 DATA 17,0,175,205,129,98,205,62,98
,58,169,97,254,3,40
110 DATA 11,62,3,50,169,97,33,128,0,34
,184,97,33,156,99
120 DATA 34,162,97,201,205,41,98,221,
33,156,99,237,91,180,97
130 DATA 62,255,205,194,4,6,50,116,16,
253,217,33,88,39,217,201
140 CLEAR 30000: FOR I=30001 TO 30196:
READ A: POKE I,A: NEXT I
150 SAVE "Breaker3.2" CODE 30001,196
160 CLS: PRINT AT 10,10; FLASH 1;"VERI
FY"
170 VERIFY "Breaker3.2" CODE 30001,196
```

```
10 CLEAR 25034
20 LOAD "Breaker3.2" CODE 25035,196
25 CLS: PRINT AT 10,1;"Snimi odmah iza
BASIC PROGRAMA"
30 SAVE "Breaker3.2" CODE 25035,196
35 CLS: PRINT AT 10,10; FLASH 1;"VERIF
Y"
40 VERIFY "Breaker3.2" CODE 25035,196
50 CLS: PRINT AT 10,10; FLASH 1;"KRAJ"
```

ru dela programa koji se zove zaglavlje (,,header").

Promena zaglavlja

Program se kod Spectruma sastoji iz dva dela zaglavlja i podataka Zaglavlje zauzima 17 bajtova koji su podeljeni u nekoliko grupa Prvi bajt daje računaru informaciju o tipu programa (OO za bejzik, 01 za brojni niz, 02 za alfanumerički niz i 03 za ono što nas najviše interesuju mašinski program snimljen sa CODE ili SCREEN), sledećih deset bajtova sadrže ima programe (ako ime ima manje od 10 simbola, praznine se popunjavaju blankovima, ako program nema ime, prvi od ovih deset bajtova sadrži FF), a zatim slede parametri bloka kojima treba posvetiti posebnu pažnju

Prva dva parametra predstavljaju dužinu bloka u bajtovima (ako sa B_x obeležimo sadržaj x-tog bajta zaglavlja, dužina programe se računa kao 256 $B_{12}+B_{11}$), a sadržina sledećih četiri zavisi od tipa programa

Ako se radi o BASIC programu, broj linije od koje se program automatski startuje je dat sa $B_{13}+256\ B_{14}$ Obzirom da B_{12} i B_{11} daju dužini programa zajedno sa promenljivima, potrebni su bajtovi 15 i 16 za ,,pamćenje" dužine samog programa. Dužina se kao i obično, izračunava kao $B_{16}\ 256+B_{15}$.

Ako je sniman numarički niz, B_{14} sadrži njegovo ime (kod Spectruma imena nizova mogu da imaju samo jedno slovo) pri čemu je njegov bit 6 uvek nula Ako je sniman alfenumerički niz, njegovo ima je ponovo sadržano u četrnaestom bajtu s tim što je šesti bit sada jedan

Ako je, najzad, sniman mašinski program, apsolutna adresa od koja program trebe da bude unesen u memoriju može da se dobije kao $B_{13}+256\ B_{14}$, dok sadržaji B_{15} i B_{16} nisu bitni

Kada smo razumeli strukturu zaglavlja, možemo da upoznamo opcije koje omogućavaju manipulacije sa njim Sa **lh** učitavamo samo zaglavlje i smeštamo ga u memoriju Zatim možemo, se **ah**, da snimamo zaglavlje na novu traku, da učitamo blok koji se nalazi iza zaglavlja sa **lb** i da ga snimamo se **sb** (svako snimanje moža da prati verifikacija uz korišćenje opcije **vh** odnosno **vb**), ali ovakva manipulacija nema velikog smisla — bilo je jednostavnije da smo koristili opciju **bp.** Posle unošenja zaglavlja, međutim, možemo da izaberemo opciju **th** i računar će ispisati sadržaj zaglavlja i omogućiti nam da ga menjamo Možemo, na primer, da izmenimo ime programa i da zatim snimimo zaglavlje i blok Mnogo je interesantnije eksperimentisati sa promenama ulaznih adresa i drugih parametara bloka, ali ovakva promena nosi ogroman rizik — možemo da zadamo parametre koji nemaju smisla i tako onemogućimo unošenje presnimljenog programa u memoriju

Programi bez zaglavlja

Ovaj ,,brejker" omogućava presnimavanja najvećeg broja (nekih 97%) komercijalnih programa Ipak, pred jednom zaštitom on je nemoćan neki autori prave programe u kojima je simulirana naredba LOAD, tako da blok sa mašinskim programom ne sadrži zaglavlje Jedini programi zaštićeni na ovaj način koji su nam došli do ruke sa fort cyrusis chesse i turk chess Preporučujemo vam da, uz pomoć disasemblera, pronađete deo mašinskog programa koji vrši snimanje i da iz njega pročitate potrebne adrese Za ovakvo proučavanje potreban je određeni nivo poznavanja Spectruma, ali vas u toku rada može držati ideja da ćete dok ne presnimite program, mnogo naučiti i da sigurno ne postoji savršen metod zaštite programa!

zagonetke sharpa PC 1500

Ima računara kod kojih je mašinsko programiranje moguće ali nedovoljno dokumentovano. Dobar primer takvog računara je PC1500: njegovi proizvođači nisu objavili nikakvu literaturu o mašinskom programiranju, pa čak nisu spomenuli ni postojeće naredbe PEEK, POKE, CALL u okviru uputstva za upotrebu! Pisma proizvođača su nas uverila da „Sharp" nema nameru da otkriva tajne mašinskog programiranja i da je na korisnicima da u tome uspeju. Ponovo NOMAS!



Zagonetka za programere Sharp PC-1500

Ne bi bilo problema kada bi Sharp PC1500 koristio procesor čiji je sat instrukcija poznat PC1500, međutim, koristi specijalizovani procesor kod koga se kodiranje instrukcija razlikuje od većine poznatih. Računar, pored toga, ima prilično čudan sistem adresiranja koji je omogućio memoriju od 128 kilobajta. Ipak, vredni francuski i američki korisnici PC1500 uspeli su da udu u njegova tajne i objave set instrukcija (iako nekompletan) kao i mapu bejzika koji se koristi. Na taj način je mašinsko programiranje postalo relativno dostupno širem krugu korisnika.

Pre nego što navedemo spisak mašinskih naredbi za PC1500, moramo da izložimo bejzik naredbe za rad sa programima na mašinskom jeziku. Da bismo uopšte mogli da unosimo mašinski program, moramo za njega da rezervišemo odgovarajući prostor. To postižemo pomoću naredne NEW. U uputstvu za korišćenje je spomenut samo njen oblik NEW O kojim se briše čitava programska memorija računara. Mi ćemo, na primer, da koristimo NEW 14000 što znači da smo 14000 bajtova memorije rezervisali za BASIC program, a preostalih 14533 — 14000 = 533 bajta za mašinska programa. Ovakva raspodela važi sva dok je ne opozovemo sa NEW 15433 (svi ovi podaci se odnose na PC1500 koji je proširen maksimalnim memorijskim modulom od 8 kilobajta, ukoliko njega nema, sve brojeve traba srazmerno umanjiti).

Za unošenje mašinskih programa, kao i uvek, koristimo naredbu POKE. Ukoliko, na primer, želimo da u memorijske ćelije čije su adrese 14500 i 14501 postavimo mašinski program kodiran kao OA 9A, koristimo POKE 14500,&OA,&9A. Za proveravenja upisanog programa možemo da koristimo PEEK (PEEK 14500 traba na ekranu da dâ 10, što je pravod broja OA u dekadni sistem).

Mašinski programi se snimaju na kasetu naredbom SCAVE M, a čitaju naredbom CLOAD M. Opšti format naredbe SCAVE je, koristeći uobičajenu Bekusovu notaciju, SCAVE M „ime", početna adresa [,krajnja adresa] a naredba CLOAD M: CLOAD M „ime" [, početna adresa]

Do sada tajanstvene naredbe operativnog sistema POKE ⋕ i PEEK ⋕ se koriste za smeštanje sadržaja u „gornjih" 64 kilobajta memorije i deluju na isti način kao POKE i PEEK koja smeštaju podatke u „donjih" 64 K. Jasno je da računar, u sadašnjim opremama, ima daleko ispod 64 K memorije, pa naredba PEEK ⋕ i POKE ⋕ izgledaju nepotrebno. Ipak, može da se desi da Sharp pusti u prodaju aplikacione ili čak memorijska module koji bi bitno proširili memoriju PC1500!

Za one koji ne žele da pišu mašinska programe ali nameravaju da upotrebe naredba PEEK i POKE, vredi reći da je kodiranje bejzik programa sada razjašnjeno. Svaka linija bejzika sadrži najpre linijski broj, zatim adresu početka sledeće linije i, na kraju, sadržaj kodiran tako da svaka bejzik naredbe zauzima dva bajta (F1 A8 je, na primer, naredbe RANDOM). Na kraju linije uvek se nalazi kod OD koji je odvaja od sledeće. Znajući ovo, moguće je sastaviti bejzik program koji vrši pranumeraciju drugih bejzik programa tako da se linijski brojevi razlikuju za po 10, pa čak i program koji bi pisao druge programe (na primer, komponovao muziku).

Mnogo je interesantnija pisanje programa na pravom mašinskom jeziku. Njih unosimo pomoću POKE, a startujemo novom naredbom CALL. Ovu naredbu prati adresni deo koji predstavlja adresu prve mašinske instrukcije koju traba izvršiti. CALL se, jasno, može naći i u samom bejzik programu.

Set instrukcija procesora LH5801 je dat u sledeće dve tabela. Prva predstavlja funkcije standardnih naredbi, dok druge daje instrukcije koje se izvršavaju kada se ispred nekog koda nađe FD. Ovaj kod je neka vrsta prefiksa koji je stvorio mogućnost proširenja seta instrukcije sa 256 novih, čime je broj naredbi povećan na 511 bez ikakvog utroška memorije. U tabelama (posebno u drugoj) ima dosta praznih mesta — to su instrukcija čije značenje još nije poznato. Otkrivena instrukcija daju mogućnost pisanja veoma složenih programa, ali nedostatak poznavanja svih instrukcija onemogućava disasembliranje ROM-a u kome ih ima dosta (nova formulacija Marfijevog zakona: ako ne poznajete značenje jedne naredbe, onda se ona najčešće javlja u ROM-u).

U tabelama su instrukcija objašnjene korišćenjem standardne nomenklature procesora Z80. Ipak, Z80 ima mnogo manje naredbi od LH5801, pa može da se desi da manje iskusan korisnik ne može da shvati funkciju neke naredbe poznajući samo njenu skraćenicu. Prostor u ovom izdanju nam, na žalost, ne omogućava da objasnimo sve naredbe, pa nam samo ostaje da obećamo da će se u Katalogu našeg Kluba pojaviti program-uputstvo broj PC1500901300 koje će biti mnogo opširnije. U međuvremenu, navodimo primer programa koji demonstrira brojanje od 0

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	SBC A,C	SBC A E	SBC A,L		INC C	INC E	INC L		SBC A B	SBC A D	SBC A H		SBR C0	SBR D0	SBR E0	SBR F0
1	SBC A (BC)	SBC A (DE)	SBC A (HL)		LDI (BC) A	LDI (DE),A	LDI (HL),A		JR NC +e	JR NC −e	SBC A,(nn)	SBC A,n	SBR NC n	RRA		REX
2	ADC A,C	ADC A E	ADC A,L		DEC C	DEC E	DEC L		ADC A B	ADC A D	ADC A H		SBR C2	SBR D2	SBR E2	SBR F2
3	ADC A (BC)	ADC A (DE)	ADC A (HL)		LDD (BC) A	LDD (DE) A	LDD (HL) A		JR C +e	JR C −e	ADC A,(nn)	ADC A,n	SBR C n	RRD		
4	LD A C	LD A E	LD A,L		INC BC	INC DE	INC HL		LD A,B	LD A D	LD A H		SBR C4	SBR D4	SBR E4	SBR F4
5	LD A (BC)	LD A,(DE)	LD A (HL)		LDI A (BC)	LDI A (DE)	LDI A,(HL)		JR NH +e	JR NH −e	LD A (nn)	LD A n	SBR NH n	SRL		LDI (DE), (BC)
6	CP A C	CP A E	CP A L		DEC BC	DEC DE	DEC HL		CP A,B	CP A,D	CP A H		SBR C6	SBR D6	SBR E6	SBR F6
7	CP A (BC)	CP A (DE)	CP A (HL)		LDD A,(BC)	LDD A (DE)	LDD A (HL)		JR H +e	JR H −e	CP A (nn)	CP A n	SBR H n	RLD		
8	LD B,A	LD D A	LD H A		LD B n	LD D n	LD H,n		DJC −e				SBR C8	SBR D8	SBR E8	SBR F8
9	AND A,(BC)	AND A (DE)	AND A (HL)		AND (BC) n	AND (DE) n	AND (HL) n		JR NZ +e	JR NZ −e	AND A (nn)	AND A n	SBR NZ n	SLA	AND (pp) n	RCF
A	LD C,A	LD E,A	LD L,A		LD C n	LD E n	LD L,n			RET	LD SP nn	JP nn	SBR CA	SBR DA	SBR EA	SBR FA
B	OR A (BC)	OR A (DE)	OR A,(HL)		OR (BC) n	OR (DE),n	OR (HL) n		JR Z +e	JR Z −e	OR A (nn)	OR A,n	SBR Z n	RLA	OR (pp) n	SCF
C	SBCD A (BC)	SBCD A (DE)	SBCD A (HL)		CP B n	CP D n	CP H n		ADCD A (BC)	ADCD A (DE)	ADCD A (HL)		SBR CC	SBR DC	SBR EC	SBR FC
D	XOR A (BC)	XOR A (DE)	XOR A,(HL)		TEST (BC) n	TEST (DE) n	TEST (HL) n		JR HV +e	JR NV −e	XOR A (nn)	XOR A,n	SBR n	INC A	TEST (pp) n	2nd table
E	LD (BC),A	LD (DE) A	LD (HL),A		CP C n	CP E n	CP L n		JR +e	JR −e	LD (nn) A	CALL nn	SBR CE	SBR DE	SBR EE	SBR FE
F	TEST A (BC)	TEST A (DE)	TEST A (HL)		ADD (BC) n	ADD (DE) n	ADD (HL) n		JR V +e	JR V −e	TEST A (nn)	TEST A,n		DEC A	ADD (pp) n	

256 256=65536 i, uzgred, prednost mašinskog jezika na BASIC-om.

```
4100 4A    00     LD R1,0
4102 B5    00     LD A, O
4104 DD           INC A
4105 99    03     JR NZ - 3
4107 40           INC R1
4108 99    08     JR NZ - 8
410A 9A           RET
```

Za unošenje ovog programa najpre rezervišite potreban prostor sa NEW &4200 a onda ga unesite pomoću POKE&4100,&4A, 0,&B5, 0,&DD, &99,3 &40,&99,9,&9A Izvršavanje programa se postiže sa CALL &4100 Ne vredi da uključujete štopericu — nećete stići ni da primatite kada je program završio sa radom Obzirom da „brojanje" traje manje od jedne sakunde, mnogi vlasnici PC1500 ne veruju da je program dobro napisan! Bejzik program bi isti problem rešio za petnaestak minuta

KLUBOVI PROGRAMERA

Kada proučite ovu školu bejzika i počnete da sastavljate samostalne programe poželećete da postanete član nekog Kluba programera i da razmenjujete programe i iskustva sa drugim vlasnicima računara Klubovi programera u svetu su obično specijalizovani za pojedine računare, dok jugoslovenski mahom tretiraju sve modele Evo najpre nekoliko inostranih adresa

Američki PPC klub (2545 W Camden Place, Santa Ana CA 92704, USA) je najstariji svetski klub vlasnika stonih i džepnih računara Uglavnom je orijentisan na proizvode firme Hewlett-Packard i izdaje dva izvanredna časopisa PPC Calculator Journal (posvećen džepnim) i PPC Computer Journal (posvećen stonim računarima) Časopisi izlaze 10 puta godišnje, a članarina, u koju je uračunata i pretplata, iznosi 37 dolara za svaku od aktivnosti (kalkulatori ili kompjuteri)

Trenutno najveći klub korisnika nekog stonog računara je BEEBUG, P O Box 109, Baker Street (broj nije 221 b), High Wycombe, Bucks HP11 2TD, England Klub se bavi BBC mikroračunarima (naročito modelom B) i u poslednje vreme Atom Electronom Za godišnju članarinu od 16 funti dobija se 12 brojeva časopise od kojih svaki ima 64 strane — bez reklama

Najveći klub korsnika Sinclairovih računara je koncentrisan oko ECC Publications, 30 Islington Green London NI 8BJ England Oni koji se odluče da otkucaju Breaker 3 2 i tako steknu mogućnost da presnimavaju zaštićene programe mogu bitno da povećaju svoju zbirku programa ako se učlane u Sinclair Owners' Software Library, Warren Road, Liss Hants GU33 7DD Za godišnju članarinu od 8 5 odnosno 9 5 funti, članovi mogu da iznajmljuju komercijalne programe na kratko vreme uz minimalnu naknadu

„Galaksijin" klub programera, o kome ćete pročitati još ponešto u ovom specijalnom izdanju, predstavlja najstariji i trenutno najaktivniji klub u Jugoslaviji Lepu tradiciju ima i riječki klub FRAK, Luje Adamiča 1/II FRAK je klub studenata Građevinskog fakulteta u Rijeci koji izdaje jedini domaći kompjuterski časopis Ovaj časopis izlazi četiri puta godišnje i objavljuje napise vezane uglavnom za građevinu Obzirom da je članarina samo 500 dinara (250 za studente) časopis će verovatno biti interesantan i za one koji se ne bave građevinarstvom U Beogradu radi SRK, klub studenata Građevinskog fakulteta U njihovom katalogu nalazi se pristojan broj isključivo stručnih programa uputstava za upotrebu TI59 i HP41C i drugo Klub se sastaje četvrtkom od 13 časova u sali 337 (Bulevar revolucije 73)

Beogradski Elektrotehnički fakultet takođe ima Klub programera koji je orijentisan i prema članovima koje elektrotehnika ne interesuje U njihovom Katalogu sada ima 150 programa za džepne računare i Sinclairove modele koji su članovima na raspolaganju uz veoma umerene naknade ili putem razmene Sastanci (često su ti sestanci pretvoreni u predavanja) održavanju se ponedeljkom u 19 časova u sali 59 Elektrotehničkog fakulteta Zainteresovani van Beograda mogu da pišu na adresu Klub programera, OSSO ETF, Bulevar revolucije 73, Beograd

U RK Nikola Tesla (Timočka 18 Beograd) zainteresovani imaju priliku da rade sa stonim računarom TRS 80 Slična mogućnost je ostavljena i onima koji žele da upoznaju Sinclairove računare — o njima se brine RK Avala, Ćirila i Metodija 2a.

Problem

Problem koji vam postavljamo na prvi pogled izgleda sasvom besmisleno, ali će se oni koji pokušaju da ga reše uveriti da je u pitanju prilično složena logičko/programerska zagonetka. Treba, naime, sastaviti program koji lista samog sebe!

Da bi program bio ispravan naredba RUN treba da dâ isti rezultat kao i naredba LIST U programu smeju da se koriste samo standardne bejzik naredbe LIST, PEEK, POKE, USR, CALL i slične su zabranjene Izuzetno, kod računara „galaksija" može da se koristi naredba BYTE Pozivamo čitaoce ovog specijalnog izdanja da nam pošalju rešenja problema — nagradićemo najkraće



IVO LOLA RIBAR
FABRIKA ALATNIH MAŠINA

lola 8



„Lola 8", jedan od prvih domaćih kućnih računara, predstavlja izvanrednu sintezu velikog iskustva SOUR „Ivo Lola Ribar" u oblasti automatskog upravljanja alatnim mašinama i mladalačke svežine tima nedavno diplomiranih stručnjaka. U toku njegovog razvoja nije se mnogo gledalo na radno vreme, podelu posla, nagrade... Mladi inženjeri su jednostavno želeli da njihov kompjuter dođe u što više domova i škola i da tako što veći broj ljudi zavoli ono čime se oni već godinama bave. Njihov rad je dao i prve rezultate: u toku ovogodišnjeg Međunarodnog sajma tehnike održano je prvo takmičenje učenika srednjih škola SR Srbije u sastavljanju programa za „lolu 8". Kao rezultat toga, „lola 8" je postao među učenicima najomiljeniji deo inventara par obrazovnih ustanova, što će svakako obogatiti njegovu biblioteku programa. Ostaje nam samo da poželimo da „lola 8" u skoroj budućnosti dođe u maloprodaju i da vas upoznamo sa njegovim karakteristikama.

„Lola 8" je, kao i većina današnjih komjutera, koncipiran tako da u osnovnoj varijanti odgovara onima koji čine prve korake u programiranju, a da sa dodavanjem posebnih modula prilagođava iskusnijim korisnicima koji imaju veće potrebe. Osnovu računara čini Intelov procesor 8085A koji radi na osnovnoj učestanosti od 5 MHz. Visoka frekvencija, uz hardver koji je oslobodio procesor svih briga za generisanje slike, čini da se čak i bejzik programi izvršavaju veoma brzo — o mašinskim da i ne govorimo.

U osnovnoj verziji „lola 8" ima ROM od 12 Kb koji može da se proširi do 18 Kb, RAM od 6 Kb se, dodavanjem posebnih modula, može proširiti do 36 Kb, što bi trebalo da bude dovoljno za sve primene. Alfanumerička tastatura sa 48 tastera sadrži, pored uobičajenih velikih slova, brojeva i specijalnih znakova, određen broj specijalnih grafičkih simbola (znakovi na kartama, na primer) i latinična slova Č, Ć, Ž i Š.

U ROM je smešten bejzik interpretator značajnih mogućnosti: pored uobičajenih osnovnih operacija sa brojevima u pokretnom zarezu, omogućeno je izračunavanje trigonometrijskih i logaritamskih funkcija sa devet tačnih cifara, rad sa matricama sa više dimenzija i crtanja slika na ekranu u rezoluciji 75·80 tačaka. Umesto skupog video terminala, „lola 8" se povezuje sa standardnim crno-belim televizorom (UHF, 36. kanal) na čijem ekranu prikazuje po 40 karaktera u svakom od 25 redova. Editor teksta omogućava jednostavno i brzo pisanje i ispravljanje programa dodavanjem ili brisanjem grupa karaktera ili čitavih delova programa.

Iskusnijim programerima i onima kojima je bejzik nedovoljno brz naredbe PEEK, poke i USR otvaraju vrata mašinskog programiranja. Ono biva značajno olakšano kada se ROM proširi 4 Kb dugačkim programom koji obavlja funkcije mini-asemblera i monitora.

U osnovnu konfiguraciju je uključen i generator zvuka koji omogućava nezavisno programiranje tri zvučna kanala za generisanje tonova u čitavom čulnom opsegu (8 oktava), uz doziranje šumova raznih amplituda i oblika. Mogućnost programskog regulisanja jačine generisanog zvuka omogućava zvučne efekte kao što je prasak, zvižduk, šum talasa itd.

Računar je snabdeven interfejsom za povezivanje sa standardnim kasetofonom na koji se brzo i lako snimaju upisan bejzik i mašinski programi koji tako bivaju sačuvani za buduću upotrebu. Čitav ovaj hardver je smešten na profesionalnoj dvoslojnoj štampanoj pločici.

Na poleđini računara se nalazi 64-pinski EURO konektor na koji se prikjučuje jedan modul ili kartica-adapter za prikijučivanje više modula. Pored palica za igre i memorijskih modula, na raspolaganju su serijski i paralelni interfeja za povezivanja računara sa mernim instrumentima. Ovi moduli se već koriste u „lolinim" laboratorijama za testiranje raznih uređaja.

Programska podrška računara je čak i u sadašnjoj fazi pristojna: „Iola 8" nudi mnogo časova zabave kroz igre Black Jack, Mastermind, Glinena flota, Melodika i druge. Pisanje „ozbiljnih" programa, kao što su interpretatori i kompajleri drugih programskih jezika, biće olakšano zahvaljujući koncepciji operativnog sistema računara „lola 8" i maksimalno podržano od firme.

Konstruktori „lole 8" će biti zahvalni za sve primedbe i sugestije koje upute čitaoci ovih redova. Na taj način će računar poprimati konačnu koncepciju koja će zadovoljiti najveći broj njegovih budućih korisnika. **Adresa na koju možete da pišete za sve informacije je „Ivo Lola Ribar", 11250 Železnik.**

8/ govorite li bejzik?

Programiranje za početnike

Ovo poglavije predstavija osnovnu školu najčešće korišćenog programskog jezika — bejzika — kojim govore svi kućni računari. Namenjeno je prvenstveno onima kojima je ovo prvi praktični dodir sa programiranjem što, naravno, ne znači da i oni koji poznaju bejzik neće iz njega moći nešto da nauče. Iako računar nije neophodan za razumevanje bejzika, nije loše imati neki pri ruci. Prateći ovu školu zasigurno ćete naići na neke sekvence i instrukcije koje će u vama probuditi dileme: zašto je nešto rešeno „ovako“ kada se moglo lakše rešiti „onako“. Najbolji način da rešite sve probleme tog tipa (i uverite se da se „onako“ nešto moglo bolje ili lošije uraditi) je da da na računaru isprobate sve varijante. Na taj način ćete najpre naučiti programiranje: pokušajte da pišete programe ma koliko pri tome grešili. Računaru vaše greška neće smetati — za njega je izvršavanje lačnog i pogrešnog programa isti posao. U programiranju se, više nego igde drugde, uči na greškama. Na sreću, ovde greške ne izazivaju nikakvu štetu!

Većina stonih računara (osim onih najskupljih) povezuje se sa standardnim televizorom ili monitorom Na samom računaru uočavate jedino tastaturu koja može biti mehanička (kod skupljih modela) ili senzoraka (kod jeftinijih računara) Senzorsku tastaturu ćete prepoznati po tome što se tasteri na kraču ili se neznatno pomeraju. Sa desne ili zadnje strane računara se, obično, nalazi nekoliko priključaka Jedan od njih je namenjen napajanju — u njega se priključuje ispravljač koji ste dobili uz računar. Jedan priključak služi za povezivanja računara sa televizorom uz pomoć posebnog kabla, koji je takođe deo opreme koju ste dobili uz računar

Uključiti, ali kako?

Neki računari će početi da rade čim ih povežete sa napajanjem (preko ispravljača), dok drugi imaju poseban prekidač za uključivanje Iako ste računar povezali sa televizorom i uključili mu napajanje, na ekranu ćete, verovatno, videti samo "sneg". Da biste dobili uobučajeno READY (znak da je računar spreman da prima vaše naredbe), potrebno je da izaberete odgovarajući TV kanal Za ovo će verovatno biti potrebna konsultacije sa vašim uputstvom za upotrebu računara. TV kanal se nalazi negde na UHF području, (kod Sinklerovih modela kanal 36) ali varira od računara do računara Kada se na ekranu pojavi READY (kod Spectruma zaštitni znak u donjem delu ekrana), možete da prebirate po tastaturi.



Pored priključaka za povezivanje sa mrežom i televizorom, na računaru se nalaze dva ili tri priključka za povezivanje sa kasetofonom Uz računar ste dobili i odgovarajući kabl To je obično pedesetak centimetara dugačak kabl se po dva utikača na svakom kraju Jedan od njih se uključuje u utičnicu kasetofona ispod koje piše MIC, a drugi u utičnicu ispod koje piše EAR (ove oznaka mogu da budu različite od kasetofona do kasetofona prva označava mikrofon a druga spoljni zvučnik ili slušalicu). Drugi par utikača se povezuje sa odgovarajućim utičnicama na računaru, pri čemu treba paziti da se na pogreši — ukoliko je jedan od crvenih utikača povezan sa utičnicom kasetofona ispod koje piše MIC, drugi crveni džek treba da bude povezan se utičnicom na računaru ispod koje se nalazi isti napis. Bolji



računari imaju i treći par utikača koje ćete poznati po tome što su nešto uži od preostala dva para. Ovi džekovi se priključuju u utičnice ispod kojih piše „REMOTE" i omogućavaju računaru da kontroliše uključivanja i isključivanje kasetofona.

Demonstraciona kaseta

Uz računar ste, eventualno, dobili kasetu sa demonstracionim programima Iako sami još ne umete da pišete programe, možete se zabavljati sa programima koje su drugi sastavili i tako upoznati neke mogućnosti vašeg računara Da unesete takav program u računar, premotajte demonstracionu kasetu na početak, otkucajte CLOAD (LOAD bi mogio da se prevede sa "unesi", a C označava da se unosi sa kasete, kod nekih računara ova naredbe može da bude i drukčija, na primer LOAD kod Sinklerovih modela) i pritisnite taster na koma piše ENTER (ili NEWLINE ili RETURN, ovaj taster ćete prepoznati obzirom da se, kod svih računara, razlikuje od svih ostalih po veličini ili dizajnu, više o ovom tasteru ćemo govoriti docnije) Zatim pritisnite taster PLAY na kasetofonu Zavisno od računara u toku unošenja programa dešavaće se različite stvari Svaki računar nekim promenama na ekranu signalizira da se program unosi, ukoliko je sve na ekranu mirno verovatno niste podesili dovoljnu jačinu zvuka na kasetofonu — pokušajte da je povećate Isključite i ponovo uključite računar, premotajte traku na početak i ponovite čitav postupak. Posle nekoliko pokušaja (a broj

CLOAD "PROKI"

Možda ćete videti isprekidane linije na ekranu dok se program učitava u računar.

Pokušajte da slušate kasetu na uobičajen način da čujete kako "program zvuči".

pokušaja je obično obrnuto srazmeran ceni računara i kasetofona), računar će ispravno unati program i na ekranu ispisati READY. Zaustavite kasetofon i otkucajte RUN pa ponovo pritisnite ENTER. Od tog momenta računar će početi da izvršava program i mi ne možemo da kažemo šta će se desiti — verovatno će na ekranu biti ispisana neka uputstva za upotrebu programa i korisniku će biti naloženo da pritiska neke tastere za dalje uputstva ili za početak rada

Prvo snimanje . . .

Prateći našu školu, napisaćete i neke programe i poželeti da ih snimite na kasetu da biste docnije ponovo radili sa njima (ili, kako programeri kažu, na njima — poslednja fraza podrazumeva rad na daljem poboljšanju programa). Da biste snimili program na traku, uzmite praznu kasetu, stavite je u kasetofon i pritisnite tastere za snimanje na njemu. Zatim otkucajte SAVE, pritisnite taster na kome su nacrtani znaci navoda (na ekranu će pisati SAVE"), a zatim otkucajte ime programa. Neki računari dopuštaju imena programa od samo jednog slova, dok se kod drugih, kao imena, mogu koristiti čitave rečenica. To nije mnogo važno — najbolje je programima davati kratka imena koja se lako pamte. Pošto ste otkucali ime programa, pritisnite ponovo taster sa znacima navoda, a zatim ENTER.

Ponovo će sadržaj ekrana početi dinamično da se menja, a snimanje vaših programa (ako ste još početnik) će biti vrlo kratko, baš kao i sami programi. Ukoliko se radi o značajnom programu, možete da ga snimite dva puta kako biste bili sigurni da neće nastupiti problemi sa njegovim učitavanjem

. . . i kucanje programa

Ako ste pretplaćeni na neki kompjutarski časopis, u njemu ćete često nalaziti interesantne programe. Kada poželite da otkucate neki od njih, pre svega proverite da li je taj program pišen za vaš računar i da li je za njegovu upotrebu potrebna neka periferijska oprema koju vi nemate. Ako program nije pisan za vaš računar najbolje da, bar za prvo vreme, odustanete od njegovog unošenja. Prepravka programa nekada može da bude zametniji posao od pisanja novog programa, jer se treba uklopiti u logiku drugog programera. Nešto više o ovoj temi govorićemo u poglavlju koje je namenjeno iskusnijim programerima

Kada pronađete program za vaš računar koji želite da otkucate moraćete malo da razmislite o načinu na koji se vašem računaru saopštavaju naredbe. Program treba uvek unositi bez ikakvih promena — ukoliko vam nešto nije jasno, ne razmišljajte mnogo o tome. Program objavljen u časopisu je svakako više puta testiran i u njemu nema grešaka (osim, možda, štamparskih, koje će svakako biti ispravljene u sledećem broju). Svaka linija ima svoj broj koji treba otkucati. Pošto ste otkucali linijski broj, unosite sadržaj same linije. Kod nekih računara morate da kucate sva slova naredbi (na primer, naredbu PRINT kucate slovo po slovo), dok drugi imaju tzv "dodeljene naredbe" — da biste otkucali PRINT dovoljno je da pritisnete taster P u pravom trenutku. Najbolji su oni računari koji omogućavaju da se naredbe kucaju slovo po slovo ili po želji korisnika, da se koriste tasteri kojima su dodeljene neke funkcije

Na nekim tasterima je napisano više simbola a izbor pravog se postiže pomoću posebnog tastera na kome piše SHIFT. Neki računari imaju nekoliko SHIFT tastera koji su označeni raznim bojama, dok svaki drugi taster ima i po tri-četiri funkcije koje zavise od toga koji je od SHIFT tastera bio pritisnut. Kucanje programa na ovakvim računarima je u početku vrlo naporno jer je potrebno mnogo traženja po tastaturi, ali se korisnik docnije, navikne na njihov raspored

Iako se može kupiti mnogo knjiga sa gotovim programima i još mnogo više programa na kasetama, pre ili kasnije svaki korisnik računara poželi da napiše neki "svoj" program. To je kreativan posao koji zahteva određeni nivo poznavanja osobine računara i programskog jezika na kom se program piše (u našem slučaju programski jezik je bejzik). Iako je za pisanje programa potrebno utrošiti određeno vreme, sam posao je zabavan i dinamičan i često predstavlja bolju zabavu od igranja sa svim kupljenim kompjuterskim igrama. Sledeće stranice su namenjene onima koji žele da napišu svoj prvi program



Prve reči

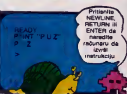
Kao što znamo, čitav "razgovor" sa računarom se obavlja preko tastature. Na njoj korisnik slovo po slovo, kuca naredbe koje su, jednostavno, kratke rečenice na engleskom koje govore kompjuteru šta treba da uradi. Kao što se u običnom govoru na kraju svake rečenice stavlja tačka, tako se u bejziku, na kraju naredbe, pritiska poseban taster koji je (kao što smo rekli) označen sa ENTER, NEWLINE, RETURN, EXECUTE ili nekim drugim napisom (mi ćemo u daljem tekstu koristiti oznaku ENTER pošto se javlja kod najvećeg broja računara). Pritisak na ENTER (=ubaci, izvrši instrukciju) govori računaru da je korisnik otkucao ono što je imao da kaže i da je sada na kompjuteru da analizira i izvrši naredbu

Čak i u svakodnevnom životu svaki niz slova (pa čak ni svaki niz reči) nije rečenica. Da bi govorna rečenica bila jasna, moraju da se poštuju određene pravopisna pravila. Isto važi i za programska jezike, samo su ovde pravopisna pravila mnogo strožija. Svaka rečenica mora da bude sintaksno potpuno ispravna da bi je računar razumeo. Jedno pogrešno slovo i na ekranu će biti ispisano SYNTAX ERROR (sintaksna greška) ili samo znak pitanja pa će korisnik morati ponovo da otkuca naredbu ali ovoga puta na ispravan način. Za početnika je, kao što smo rekli, bitno da shvati da ove pogreške ne oštećuju računar, zato slobodno probajte naredbe bez obzira na česte greške



Pored READY, na ekranu se na početku reda pojavljuje simbol nazvan "kurzor". Kurzor označava mesto na kom će se pojaviti sledeći karakter koji pritisnemo. Pritisnimo, na primer slovo P. Na ekranu je prikazano to slovo, a kurzor se pomerio za jedno mesto u desno od slova P. Nastavimo da pritiskamo tastere R I N T i na ekranu će se naći prva BASIC reč koju smo "izgovorili" — PRINT. Sada pritisnimo ENTER i primetimo da se na ekranu ponovo pojavilo READY, ali pomereno dole za jedan red. Šta se dogodilo? Računar je shvatio da smo zahtevali da on nešto "štampa" na ekranu (PRINT = štampaj, ali ovde znači i "prikaži") ali, pošto nismo napisali šta treba da se štampa, prikazao nam je prazan red i dao do znanja da možemo da nastavimo





Pokušaćemo ponovo. Otkucajmo PRINT ali posle toga, otvorimo znake navoda. Znaci navoda označavaju tekst koji računar treba doslovno da prenese na ekran. Zatim pritisnimo slovo P, pa jedan razmak, pa slovo U, pa još jedan razmak i na kraju slovo Z. Obzirom da je tekst završen, pritisnemo ponovo znak navoda i ENTER. Računar nas je poslušao: na ekranu je ispisao P U Z. U prvi mah će vam izgledati teško da razumete reči u kojima su slova č, ć, š i ž (kojih nema u engleskom alfabetu) zamenjena sa c, s i z, ali ćete se ubrzo i na to navići

Pošto je sve ovako dobro prošlo, mogli bismo i da nastavimo. Otkucajmo ponovo PRINT i otvorimo znake navoda, a zatim otkucajmo XYZ pa zatvorimo znake navoda i pritisnimo ENTER. Računar izvršava još jedan naš nalog i ispisuje XYZ. Apetit počinje da nam raste — zašto računar ne bi uradio i nešto korisno? Otkucajmo PRINT "1+2−3=" pa opet pritisnimo ENTER

```
PRINT "P U Z"
P U Z
PRINT "XYZ"
XYZ
PRINT "1+2-3="
1+2-3=
PRINT "RAZOCARAN"
RAZOCARAN
```

Posle svake instrukcije pritisnite ENTER

Na ekranu piše 1+2−3= Ovo je prvo malo razočaranje koje nam je računar priredio mi smo očekivali da izračuna koliko je 1+2−3 a on je samo ponovio problem Zašto? Rekli smo da naredba PRINT računaru nalaže da ono što se nalazi između znakova navoda doslovno pranese na ekran 1+2−3= ne predstavlja aritmetički izraz to su samo alfanumerici (bojevi) kao što su bili dosadašnji P U Z i XYZ U znak protesta kucamo PRINT "RAZOCARAN" i pritiskamo ENTER Na ekranu piše samo RAZOCARAN — izgleda da naša osećanja računaru ne znače baš ništa

Da obrišete ekran otkucajte CLS ili neku odgovarajuću naredbu

Ako na opisani način nastavite da izdajete računaru naredbe ubrzo će čitav ekran biti popunjen tekstom Neki računari će tada početi da ubacuju nove redove dok će prvi redovi nestajati sa ekrana dok će drugi prijavljivati grešku U svakom slučaju, možete da obrišete čitav sadržaj ekrana tako što ćete otkucati CLS i pritisnuti ENTER

Sve naredbe koje smo do sada upotrebljavali izvršavale su se onoga momenta kada pritisnemo ENTER Kompjuter poznaje još jednu vrstu naredbi — program Programske instrukcije se prepoznaju po tome što obavezno počinju tzv linijskim brojem Linijski broj se nalazi ispred instrukcije i saopštava računaru da instrukciju koja sledi zapamti i izvrši je tek kada mu to na poseban način bude naređeno Kompjuter jasno može da zapamti veliki broj instrukcija koje se ređaju po svojim linijskim brojevima (najpre će u memoriji računara da bude smeštena instrukcija sa linijskim brojem 10 a zatim instrukcija sa brojem 20 bez obzira što su one možda otkucane obrnutim redom) Pokušajmo da napišemo program Otkucajmo 10 PRINT "/ / / / / / /" i, kao i obično pritisnimo ENTER Ništa se posebno nije dogodilo — instrukcija nije izvršena već je smeštena negde u memoriji računara Sada možemo instrukciju po instrukciju da otkucamo čitav program koji će na ekranu iscrtati stilizovano lice robota

```
10 PRINT "//////"
20 PRINT "_ _ _ _"
30 PRINT "I(   )I"
40 PRINT "I -L I"
50 PRINT "VVVVVV"
60 END
```

Linijski brojevi se najčešće razlikuju za po 10 tako da možete da dodajete naredbe bez promene brojeva

```
RUN
/ / / / / / /
I         I
I (     ) I
I — L I
V V V V V V
```

Kompjuter prikazuje tačan sadržaj koji ste otkucali između navodnika, računajući i razmake

Da bismo izvršili program, otkucajmo RUN (=izvrši) i pritisnimo ENTER Obzirom da naredba RUN nije imala linijski broj, računar će je odmah "poslušati", a to će dovesti do izvršavanja našeg programa Trenutak docnije sa ekrana će nam se smešiti stilizovani lik robota

Može da vam se desi da pri kucanju programa negde pogrešite Ukoliko primetite grešku pre nego što ste pritisnuli ENTER i tako smestili naredbu u memoriju računara, možete da se poslužite tasterom koji je označen sa RUBOUT, DELETE, ili, jednostavno . Pritisak na ovaj taster briše poslednje otkucano slovo Uzastopnim pritscima na njega možete da obrišete proizvodan broj slova i ispravite sve greške Ukoliko ste pritisnuli ENTER i tek onda primetili da postoji greška u naredbi moraćete da se poslužite drugim metodom otkucajte ponovo istu naredbu sa istim linijskim brojem ali ovoga puta bez greške Pritisnite ENTER Kako u memoriji računara ne mogu da se nalaze dve naredbe sa istim linijskim brojem prva (pogrešna) naredba će biti obrisana i zamenjena onom koju ste upravo otkucali

Skladištenje podataka

U rečnicima obično piše da je računar elektronski uređaj za obradu podataka Da bi kompjuter celishodno obrađivao informacije one moraju da mu budu saopštene na način koji će omogućiti njihovo jednostavno i brzo pronalaženje i menjanje Da bi računar mogao da pronađe traženu informaciju, ona mora pre svega da ima ime Zamislimo, na primer veliku poštu koja treba da u što kraćem roku, otpremi pisma u razne gradove Najjednostavniji način da se to uredi je formiranje pregradaka U svaki pregradak se smeštaju pošiljke upućene u neki grad (na primer, u prvi pregradak se smeštaju pisma za Beograd u drugi za Novi Sad i slično) Dežurni službenik uzima jedno po jedno pismo, čita sa omotnice mesto u koje je ono upućeno i smešta je u odgovarajući pregradak Po završetku ovakvog razvrstavanja pisma iz svakog pregratka se upućuju na odgovarajuća mesta, gde se sortiraju prema opštinama i docnije ulicama

Slična je situacija i sa memorijom računara Svakom broju ili reči koju treba zapamtiti treba da se da neko ime "imena" za brojeve su A B, C, . . . , Y, Z (ako ovo nije dovoljno mogu da se koriste i imena A1, D8, ZO i slično) a za reči — A$ B$ . . . Y$, Z$ U memoriji računara mogu istovremeno i ravnopravno da se nalaze i brojevi i reči ako postoji broj obeležen slovom A, nema nikakve smetnje da postoji i reč nazvana A$

Pošto reč "ime" ima mnogo sekundarnih značenja i nije pogodna za primenu, programeri su usvojili da se A B Z A$ nazivaju promenljive Termin je usvojen zbog toga što se sadržaji "pregradaka" neprekidno menjaju tokom razvrstavanja (ili, u konkretnom slučaju, zato što se sadržaji promenljivih menjaju u toku izvršavanja programa) Promenljive koje imenuju brojeve (A S, Z1) se nazivaju numeričke a promenljive koje imenuju reči (A$ B$) — alfanumeričke promenljive

```
10 LET A=6
20 LET B=7
30 LET C=23
40 LET D=4
```

Ovo su brojevi koje treba smestiti u memoriji računara

Ovo su imena promenljivih

„GALAKSIJIN" KLUB PROGRAMERA

Inicijativa za osnivanje Galaksijinog kluba programera data je tačno pre tri godine Ponosni smo što možemo da kažemo da se naš Klub neprekidno razvijao i stvorio impozantnu Biblioteku programa koja je na raspolaganju svim čitaocima našeg časopisa i ovog specijalnog izdanja

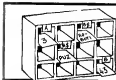
Formalna članarina ne postoji — član našeg Kluba programera je svako raspoložen za saradnju U našoj Biblioteci programa trenutno se nalazi preko 300 naslova Većina ovih programa je napisana za džepne računare TI58/59 HP41C Sharp PC1211 i PC1500, Casio FX 602P i slične Svaki od ovih programa može da se kupi za samo 40 dinara Pokazalo se da najbolju prođu imaju programi-uputstva U okviru njih je objašnjeno korišćenje nekih manje poznatih osobina računara koje u praktičnoj primeni mogu da se pokažu neobično korisnim interni registri u ubrzani režim rada za TI59 sintetičko programiranje za HP41C mašinski jezik za PC1500 Kompletan spisak programa sa njihovim apstraktima na dvadesetak kucanih strana možete da dobijete ako na žiro-račun Galaksije' uplatite 100 dinara i na našu adresu pošaljete potvrdu o uplati uz napomenu da želite Katalog broj XXXX001111 U okviru ovog kataloga ćete naći uputstvo za naručivanje programa i pročitati našu stalnu ponudu za svaki interesantan program za neki od džepnih računara koji nam pošaljete dobijate dva besplatna programa iz Kataloga po vašem izboru

U vreme pojave ovog specijalnog izdanja naš Klub programera pokreće i Katalog bejzik programa Prva počasna mesta u njemu su dobili programi-pobednici našeg nedavno okončanog Konkursa Osim njih u Katalog ćemo uvrstiti i druge domaće programe koje sastave naši čitaoci, kao i interesantne strane programe koji će se prodavati po proizvodnoj ceni koja obuhvata foto-kopiranje uputstava i snimanje programa na kasetu kao i troškove poštarine Čuveni program Hobbit za Spectruma na primer, košta svega 150 dinara Važi i razmena — za svaki vredan program koji pošaljete dobijate jedan dva ili tri programa iz Kataloga bez ikakve naknade Opširnija obaveštenja možete dobiti ako na našu adresu pošaljete kratko pismo ili dopisnicu i kažete šta vas interesuje

Članovima našeg Kluba je na raspolaganju i stalna rubrika Servis računara U njoj možete da dobijete odgovore na sva pitanja vezana za stona i džepne računare

Na početku svakog programa treba, na odgovarajući način, smestiti neophodne podatka Za to nam služi naredba LET (u bukvalnom pravodu LET znači "neka") Naredba LET A=6 ima sladaće dejatvo računar najpre oformljuje "pragradak u memoriji" koji se zove A i u njaga smešta broj 6 Slično dejstvo ima i praostalih nekoliko naredbi Po izvršavanju ovog kratkog programa raspored podataka u mamoriji će biti keo na slici

Kada smo otkucali ovaj program (na smamo da zaboravimo da pritlanamo ENTER posla svaka naredbe), poželećemo da ga izvršimo Za to će nam poslužiti već dobro poznata naredba RUN Otkucaćemo, dakla, RUN i pritisnimo ENTER Tranutak docnije na akranu će biti prikazan željeni tekat Možemo ponovo da otkucamo RUN i pritisnemo ENTER I računar će da ponovi čitav postupak Na ovom primeru se vidi zašto su programi korisni jednom smo se namučili



```
10 LET A=3
20 LET A$="P U Z"
30 LET B=43
40 LET B$="ROBOTI"
```

Ne zaboravita znakove navoda

Nekim računarima naredba LET nija potrebne, pa je LET L=O potpuno akvavilantno se L=O dok je operativni aistem drugih tako napravijen da je ova naredba neophodna Kao i uvek, ima računara koji se nalaza "negde na sredini" naredba LET im je potrabna samo u nakim slučajevima No, svi računari koji rada na bejziku imaju jednu zajedničku karakteristiku — naredba LET im nikada ne smeta Zato je za početnika dobro da je uvek koristi i teko izbegne sve potencijalna probleme

Neki računari dopuštaju da imana promenljivih budu čitava reči (npr BRZINA, CENA, DATUM i slično) No, i kod tih kompjutera na smeju da se koriste baš ave reči PRINT, RUN, LET i druge BASIC naredbe ne amaju da budu imena promenljivih da ne bi došlo do zabuna Zato korisnici računara koji imaju ovu mogućnost trebe da budu posebno pažljivi da dobro upoznaju sve bejzik naredbe ili da za imena promanljivih, koriste srpskohrvatska reči Ako ne žele da se izlažu nikakvom riziku, mogu da koriste imana promenljivih od jednog slova — i takvih, bar za početak, ima dovoljno

Situacija postaje složenija kada na odgovarajuće mesto traba da se stavi neki tekst Takst sa kao i do seda, označava navodnicima na njegovom početku i kraju, a promanljiva kojom je označen tekst mora na kraju svog imena da ima simbol za dolar Ukoliko želite da izgovorite A$ (na primer dok telefonom diktirete prijetelju neki program), recite "a dolar" ili "a string" Druga verzija nije baš u duhu našeg jezika ali je opšte prihvaćena među programerima

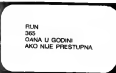
da sastavimo program (u našem slučaju to nije bilo baš mnogo teško), a zatim možemo da ga izvršavamo mnogo, mnogo puta Naravno, korist od ovog programe nije prevelika, ali na smemo da zaboravimo da pravimo tek prve koraka u bejziku

```
10 LET B=365
20 LET D$="DANA U GODINI"
30 LET L$="AKO NIJE PRESTUPNA"
40 PRINT B
50 PRINT D$
60 PRINT L$
70 END
```

mnogim kompjuterima ne traba naredba ENO



Često će nam biti potrabno da naložimo računaru da na ekranu prikaže sadržaj neke promenljive Za to, kao i do sade, koristimo naredbu PRINT ali, umsato da u njoj navodimo tekst koji traba prikazati, navešćemo ime promenljive koja sadrži taj tekst Ovaj kratki program ilustruje kako se to radi Najpre ja promenljivoj B dodeljen sadržaj 365 Obratite pažnju da smo napisali LET B=365, a ne LET B="365" Da smo uradili ovo drugo, računar bi prijevio grešku jer je B numerička promenljiva a za računar bi broj 365 bio tekat pošto se nalazi pod navodnicima

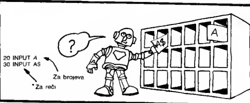


Naredba INPUT

Svi programi koje smo do sada sastavili izvršavali su se uvek na isti način, pa od njih nije bilo velike koristi Sledeća naredba koju ćemo upoznati omogućava da pri svakom izvršavanju programa damo računaru različite podatke koje će on dalje obrađivati Ta naredba je INPUT (INPUT=unesi).

Naredba INPUT mora da bude praćena imenom neke promenljive da bi računar znao kako da nazove broj (reč) koji će korisnik otkucati Naredba INPUT A (INPUT A$) na primar, izaziva privremeni prakid izvršavanja programa i pojavu upitnika na akranu Ovaj upitnik upozorava korisnika da otkuca neki broj (reč) i da, kao i obično, pritisne ENTER Računar nastavlja se radom i od tog momenta se broj (reč) koju je korisnik otkucao, u računaravoj mamoriji, naziva A (A$)

U programu može da se nađa više naredbi INPUT — prema potrabama programera Isto tako računar može više puta da izvršava naredbu INPUT A pri čemu će stari sadržaj „fahe" nazvanog A biti zaboravljan pri svakom izvršavanju ove naredbe U daljem tekstu ćemo dati nekoliko primara Ukoliko se odlučite da otkucate i izvršite neke od datih programa nema nikakve potrabe da uvek iskreno odgovarate na pitanja koja vam računar postavlja Ako svaki put kada vas računar pita za ima odgovarate na isti način naredba INPUT gubi svaki smisao (smisao naredbe INPUT je da se pri svakom izvršavanju programa računaru daju drugi podaci a svako



slika 16

ima samo jedno ima) Zato dajte mašti na volju i kucajte sva moguča imana, šala i drugo Računaru je to svejedno a on ako za to nije programiran (a sastavljanja takvih programa je i dalje van dometa našeg znanja), neće primatiti da ga „lažemo" i da sa „šalimo na njegov račun" Programiranja se najlakše uči kroz igru Ukoliko imate bilo kakve dilama o toma kako se koristi naredba INPUT ili ako su u vašem uputstvu za upotrebu računara spomanuta još naka njene osobine pokušajte sve verzije i odaberite onu koja vam je za praktičnu primenu najpogodnija





Kratak program koji je dat na slici objašnjava kako računar raaguje na naredbu INPUT Prvi upitnik sa pojavijuja na akranu kada računar naiđe na naredbu 10 INPUT G Korisnik zatim kuca naki broj koji se od tog momanta u mamoriji računara obeležava sa G Zatim sa na akranu pojavijuje novi upitnik koji je izazvan naredbom 20 INPUT B$ Ovoga puta korisnik traba da otkuca naku reč obzirom da je B$ ima za reč Važno je da primatimo da, kada treba da otkucamo reč možemo da otkucamo i broj (rekli smo da se rači sastoje od slova i brojeva) ali da na smamo da otkucamo reč kada računar očekuje broj Ukoliko pokušamo, računar će prijaviti naku grešku (najčešća TYPE MISMACH = pograšan tip promanljive)

Pošto ste otkucali program dat na ovoj slici otkucajte RUN i pritisnite ENTER Zatim najpre otkucajte vaše ima i starost ili nako šaljivo ima i nemoguću starost kao u našem primaru Izvršite program još nakoliko puta (svaki put morata da otkucata RUN i pritisnata ENTER) i svaki put unosite druge brojeve i reči Pokušajte i da na kucajući odgovor pritisnata ENTER Razni računari različito reaguju na ovu „provokaciju"!

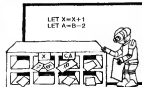
Aritmetički izrazi

Cilj aritmetičkih izraza je, kao što im i ima govori da navedu kompjuter da nešto izračuna Aritmatički izraz se sastoji od nekoliko promenljivih i konstanti koje su odvojana znakovima aritmatičkih operacija zagradama i (aventualno) nazivima specijalnih funkcija Aritmatički izraz sam za sebe nema nikakvog smisla jer računaru treba rači šta da uradi sa brojem koji je dobio izračunavanjem tog izraza Taj broj se najčešće prikazuje ili pamti koristeći neku novu promanljivu

Aritmatički izrazi se najčešće pišu ne u matematici uobičajan način uz dve razlika umesto znaka za množenja konsti se zvezdica (*) a umesto znaka za deljenja kosa crta (/) Da vidimo kako se to radi u praksi

Ukoliko nam ja potrebno da izračunamo vrednost nakog izraza nema potrabe da pišemo program Dovoljno ja da otkucamo PRINT (ne stavijajući linijski broj ispred ove naredba) i da navedamo čitav brojni izraz Na kraju, kao i obično pritiskamo ENTER i računar prikazuje vrednoat izračunatog izraza U okviru izraza mogu da se koriste i specijalna funkcije (kvadratni koran logaritam eksponencijalna funkcija, trigonomatrijske funkcija i slično) koje su nabrojane u vašem uputstvu za upotrebu računara

Početnici najčešće koriste kvadratni koren koji se u BASIC-u obalažava sa SQR Broj čiji se koren traži mora da se nađe iza SQR i to u zagradi U zagradi može da se nađe i čitav aritmatički izraz — računar ča najpre izračunati vrednost tog izraza, a zatim naći kvadratni koren te vrednosti

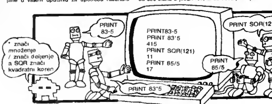


Za početnika mnogo veći problem predstavlja korišćenja naredba LET za dodaljivanja vrednosti nakog aritmetičkog izraza nakoj promanljivoj Ako na primar napišemo LET A=B−2, računar će najpre izračunati koliko je B−2 (što zavisi od vrednosti koja ja ranije dodaljena promanljivoj B) a zatim će dobijeni broj zapamtiti i u daljem radu, zvati ga A Ukoliko je, na primar, B bilo 10 (što znači da je u ranijem toku programa računar izvršio naredbu LET B=10), A će dobiti vrednost 8(10−2=8) To ja prilično jsano, ali šta sa naredbom koja kod laika predstavlja „užas" i pokazuja „koliko je programiranja neegzaktno" LET X=X+1? Besmialano je jasno, reči da je X jednako X+1 obzirom da to nije ispunjano ni za jednu vrednost X No ovu naredbu treba shvatiti drukčije računar najpre iz „faha" u memoriji koji se zove X uzima vrednost broja koji je tu „zapamćen" toj vrednosti dodaja jedan i pamti razultat u istom „fahu" gde se broj nalazio i pre Varovatno se sećate da smo rakli da promanljive manjaju vrednosti u toku izvršavanja programa Ukoliko ja X imalo vrednost 10, ono, primanom naredba LET X=X+1 dobija vrednost 11 Dakla, znak jednakosti u aritmetičkim izrazima ne treba tumačiti kao tvrdnju da su neka dva broja jednaka nego kao naredbu da se nekoj promenljivoj dodali neka vrednost Na to nas stalno podseća naredba LET iako ona, kod nekih računara nije obavezna

Još nešto o naredbi PRINT

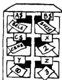
Do seda smo naučili kako da koriatimo PRINT da prikažemo brojeve reči i sadržaje nekih promenijivih Svako ko ja, čitajući dosadašnji tekst, pokušao da napravi neki samostalan program zažalio je što nema mogućnosti da ne akranu prikaže više promenijivih i tekstova u istom redu i tako programu da više dinamičnosti BASIC to omogućave — za to nam služi znak ; koji ukiapamo u naredbu PRINT

Znak ; se stavija između imana dve promanijive ili između imana promenijive i teksta koji treba da se prikažu u istom redu U avakom redu može da stane (zavisno od računara) 30—80 znakova, što znači da morate da komponujate tekst tako da oeo stana u jedan red, ili da buda podeljen u





Kod većine računara morate da ostavite prostor na svakoj strani promenljive. Prostor se stavlja pod znacima navoda




```
PRINT "UJUTRU PIJEM", X , "SOLJE", A$, "KAFE"
UJUTRU PIJEM 2 SOLJE BELE KAFE
PRINT "UJUTRU PIJEM", Z, "SOLJE", C$, "KAFE"
UJUTRU PIJEM 3 SOLJE CRNE KAFE
A$        BELE        B$ SVEZE
C$        CRNE        X 2
Y         8           Z 3
```

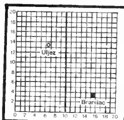
dva reda na logičan način Na slici je dat primer naredbi koje prikazuju naku vrstu jelovnika. Da bi ona mogla da se izvrše, potrebno je promenljivim dodeliti vrednosti (to radimo pomoću naredbe LET) date u levom uglu slike

Svemirska potera

Posle ovoliko učenja, red je da primanimo atečeno znanja i napišemo prvi ozbiljni bejzik program To će biti igra za dva igrača koju smo nazvali „Svemirska potera"

Pre početka igre svaki od igrača crta tablu 20 20 na listu papira sa kvadratićima Jedan od igrača predstavlja stranog uljeza, a drugi komandenta vasionskog brode koji treba da ga zarobi Svaki igrač, pre početka igre, smešta svoj svemirski brod na neko od polja table i to prvi igrač na njenoj levoj polovini a drugi igrač na desnoj Ni jeden od igrača, jasno, na zna koordinate polja njegovog protivnika Zatim igrači redom daju koordinate svoga broda računaru, a on ispisuje koliko su brodovi udaljeni (jedinica za merenja udaljenosti je jeden kvadratić) Zatim prvi igrač pomara svoj brod na neko od devet susednih polja i kuca njegove nova koordinate Isto radi i drugi igrač, a zatim kompjuter prikazuje nuvu udaljenost brodova Igra se nastavija sve dok se brodovi na nađu ne manja od 1,5 kvadratića udaljeni jeden od drugog U tom slučaju igra se prakida i smatra se da je drugi igrač pobedio Ukoliko igra potraje, na primer 60 poteza, a drugi igrač na pobedi, smatra se da je prvi pobedio.

Pri analiziranju ovoga programa obretita posebnu pažnju na naredbe čiji su linijski brojevi 50 i 100 — ove dye naredbe brišu akran računara kako sledeći igrač na bi pročitao gde je prethodni postavio svoj vasionski brod Naredba 110 je „srce" čitavog programe — ona izračuneva udaljenost brodova. Pri toma je korišćena dobro poznata formula za nalaženja rastojanja dva tačke u Dekartovom koordinatnom sistemu ($d=(x_1-x_2)^2+(y_1-y_2)^2$).



Poređenje brojeva

Igrajući „Svemirsku poteru" zapitali ste se više puta zbog čega igrači moraju da broje poteze i razmišljaju da li je rastojanje brodova manja od 1,5 kvadratić Zašto ne bi računar o tome vodio računa i, kada za to dođe vrema, sam proglasio pobednika? Da bi to postalo moguće moramo da upoznamo još dva vrste naredbi naredbe za poređenja brojeva i naredbe za ostvarivanje ciklusa

Svi programi koje smo do sada napisali izvršavali su se onim redom kojim su naredbe otkucana (uvek se prvo izvršavela naredbe 10 zatim naredba 20 i slično) U nekim slučajevima korisno je promeniti tok izvršavanja programa i preskočiti neke naredbe a neke druge izvršiti i po nekoliko puta, zavisno od ispunjenja nekih uslova Za ispitivanje ispunjanja nekih uslova služe nam naredbe IF i THEN (IF znači ***ako*** *a THEN* ***onda****).*

Na slici je prikazan najjednostavniji primer upotrebe naredbe IF Na početku programe računar pita kakvo je vrema i odgovor koji ja korisnik otkucao „pamti" pod imenom W$ Zatim nailazi na naredbu 30 u kojoj se ispisuje da li je reč koju je korisnik otkucao (i koja ja zapamćena kao W$) KISA Ukoliko je odgovor na ovo pitanja potvrdan, računar će izvršiti naredbu koja se nalazi iza THEN (u našem slučaju računar na ekranu prikazuje PONESI KISBRAN) Ukoliko je odgovor na pitanje „da li je reč koju je korisnik otkucao KISA" odrečan računar preskače ono što je napisano iza THEN naredbe i izvršava sledaču naredbu To je još jedna IF naredba pomoću koja računar ispituje da li je korisnik možda otkucao

Saveti o vremenu

```
10 PRINT "KAKVO JE VREME?"
20 INPUT W$
30 IF W$="KISA" THEN PRINT "PONESI KISOBRAN"
40 IF W$="SUNČANO" THEN PRINT "DIVNO!"
50 END
```



Ukoliko otkucate jednu od ovih reči, neće se dogoditi ništa

```
RUN
KAKVO JE VREME
?SUNČANO
DIVNO!
RUN
KAKVO JE VREME
?KIŠA
PONESI KIŠOBRAN
```

reč SUNCANO Ukoliko jeste, na akranu se prikazuje računarev komentar i program završava sa radom Ukoliko reč koju je korisnik otkucao nije ni KISA ni SUNCAND računar na izvršava ni jednu od naredbi koje sa nalazi iza THEN i jednostavno prastaje sa radom, ne prikazevši ništa

U prethodnom primaru je korišćena naredba IF kojom smo poredili da li su neka dve reči jednake Naredba IF može da se koristi i za mnogo što-šta drugo, u zavisnosti od znaka koji se nalazi između brojeva (reči) koja se poreda Ti znaci mogu da budu

nastavak na 93 strani

primer 2

```
 10 PRINT "X KOORDINATA ULJEZA?"      ove linije primaju poziciju uljeza i smeštaju
 20 INPUT A                            je u memoriju
 30 PRINT "Y KOORDINATA ULJEZA?"
 40 INPUT B

 50 CLS
 60 PRINT "X KOORDINATA BRANIOCA?"    ove linije smeštaju u memoriju poziciju
 70 INPUT C                            branioca
 80 PRINT "Y KOORDINATA BRANIOCA?"
 90 INPUT D

100 CLS
110 LET X=SQR((A-C)*(A-C)+(B-D)*(B-D))      izračunavanje rastojanja

120 PRINT "TRENUTNO RASTOJANJE JE"     štampanje izračunatog rastojanja
130 PRINT X "ZVEZDANIH JEDINICA"

140 PRINT "PRIPREMITE SE ZA DAVANJE NOVIH"   čekanje sledećeg poteza
150 PRINT "POZICIJA"
160 GO TO 10
```

mali rečnik bejzika



ABS



Izračunava apsolutnu vrednost broja ili izraza koji je . Na primer ABS) daje vrednost , a ABS 7) daje vrednost 5

A$



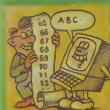
Uobičajeno ime za niz slova. Naredbe 1 INPUT A$/20 PRINT A$ na primer zahtevaju od ko ka da otkuca neku grupu slova, a zatim h rač ponavlja na ekranu. Imena za alfanumeričke promenljive su B$ C$ A9$ i slično

CALL



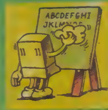
Kod nekih računara ova naredba izaziva poziv potprograma koji može da bude potpu odvojen od glavnog programa Često mogu da se prenesu i argumenti koje će računar obraditi No većina računara koji rade na bejziku ne poznaju o aredbu

CHR$



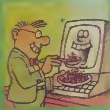
Iza ove naredbe treba da se nađe ili izraz u zagrad Računar tada generiše karakter koji je p žen navedenom broju PRINT CHR$ će većine raču ara na ekranu ispisati slovo A

CLS



Naredba shvatljiva čak i početnicima: nj izvršavanju ekran biva potpuno izbrisan

DATA



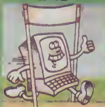
Označava početak liste podataka koje r čita kada naiđe na naredbu READ

DIM



Naredba za dimenzionisanje nizova DIM A(10) na primer nalaže računaru da rezerviše mesto u memoriji za niz koji će se zvati A i imati 10 elemenata koji će se zvati A(1) A(2), A(9) A(10) Kod većina računare na isti način može da se dimenzioniše i alfanumerički niz

END



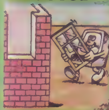
Još jedna sasvim jednostavna naredba po nailseku na nju računar prestaje se radom

FOR. . .NEXT



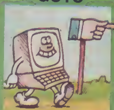
Naredba za obrazovanja ciklusa FOR I 1 TO 10 na primer označava početak ciklusa koji će da se izvršava 10 puta. Na kraju ciklusa treba da se nađe NEXT I

GOSUB



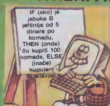
Poziv potprograma. GOSUB 1000 na primer nalaže računaru da počne da izvršava potprogram koji počinje od programske linije 1000 Računar se vraća iza GOSUB kada naiđe na naredbu RETURN

GOTO



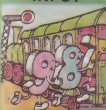
I ova naredba izaziva prelazak na liniju čiji je broj naveden (npr GOTO 100) ali je povratak na naredbu iza GOTO nemoguć Zato se GOTO naziva naredbom bezuslovnog preiseka.

IF. . .THEN. . .ELSE



Ovo su naredba uslovnog preiseka Ukoliko je uslov naveden između IF I THEN ispunjen izvršava se naredba iza THEN, ukoliko nije računar izvršava naredbu iza ELSE Primer IF A=B THEN PRINT „JEDNAKI' ELSE PRINT „RAZLIČITI'

INPUT



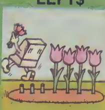
Nailazak na ovu naredbu nateruje računar da privremeno prekina izvršavanje programa i sačeka da korisnik unese broj (ili reč) koji postaje sadržaj neke promenljive Na primer INPUT A zahteva od korisnika da unese broj koji će računar od tog momenta zvati A

INT



Izračunava ceo deo izraza u zagradi Na primer INT (7/6) daje vrednost 1 jer je 7/6=1 166667

LEFT$



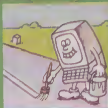
Izdvaja prvih nekoliko slova iz reči čije je ime navedeno Ako je na primer promenljivoj A$ dodeljena reč GALAKSIJA naredba LEFT$ (A$,5) daje vrednost GALAK

LET



Ova naredba je najčešće neoobavezna i služi za dodeljivanja vrednosti nekoj promanljivoj Kod većine računara LET A=5 ima isto dejstvo kao i A=5

LINE



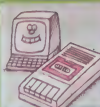
Ovu naredbu poseduju samo nešto [illegible] računari ona spaja tačke čije su koordinate navedena povlačeći pravu (ili testerastu ako je rezolucija slaba) liniju između njih

LIST



LIST je naredba koju ćete vrlo često kucati ona izaziva prikazivanja programa koji se nalazi u memoriji računara na ekranu Na ovaj način program jednostavno može da se analizira i uz primanu posebnih naredbi prepravlje Mašinski programi [illegible] ogu da se prikažu ne ekranu pomoću LIST

LOAD



Izaziva učitavanje programa se kasete Kod nekih računara je neophodno navesti i ime programa n-r LOAD "PROG"

MID$



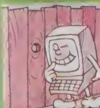
Važna naredba za red sa alfanumericima. Naredba PRINT MID$ (A$6 4) izdaje na ekranu reč SIJA ukoliko je promenljivoj A$ ranije dodaljena reč GALAKSIJA

NEW



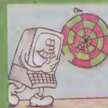
Briše program koji se nalazi u memoriji računara i priprema ga za unošenje sledećeg Kod nekih kompjutera ova naredba je ekvivalentna sa gašenjem računara dok se kod drugih čuvaju sadržaji promenljivih

PEEK



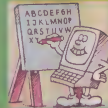
PEEK je naredba za naprednija programera PEEK (16000) na primer ispituje sadržaj bajta memorije koji se nalazi na apsolutnoj adresi 16000

POKE



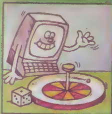
Još jedna naredba [illegible] POKE 16000 123 na primer, smešta broj 123 u memoriju i to u ćeliju čiji je broj 16000 Obzi[illegible] da se na ovaj način prethodni sadržaj ćelije 16000 briše treba biti vrlo oprazan sa POKE lako može da se dogodi da vam se BASIC program izbriše pa čak i da se računar totalno blokira

PRINT



Svakako najčešće korišćena BASIC naredba PRINT nalaže računaru da na ekranu prikaže neki komentar ili rezultat nekih računanja

RANDOMIZE



Priprema generator slučajnih brojeva za rad. Kod nekih računara ova naredba je neobavezna.

READ



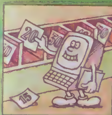
READ A [illegible] primer [illegible] pronade prvi broj [illegible] se nalazi u DA[illegible]A tabl[illegible] da ga dodeli promenljivoj A. Na isti način mogu da se [illegible]čitaju [illegible]rači

REM



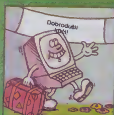
Iza REM možete da stavite proizvoljan komentar, ime programa, datum pisanja, neku opasku [illegible] REM [illegible] m[illegible]m[illegible] iako se na njih računar ne obazire pri izvršavanju programa.

RENUM



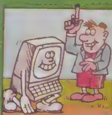
Izaziva prenumerisavanje programa [illegible] da naredbe dobiju brojeve 10, 20, 30, Mnogi raču[illegible]ri nemaju [illegible] naredbu [illegible] kod drugih se zove drukčije (npr. NAME kod TRS 80).

RETURN



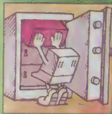
Izaziva prekid izvršavanja potprograma i povratak u glavni program iza naredbe GOSUB.

RUN



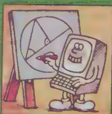
Započinja izvršavanja programa od prve linije.

SAVE



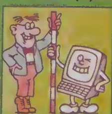
SAVE se koristi za snimanje programa na kasetu. Kod nekih računara treba navesti ime programa npr. SAVE „PROG"

SIN(A)



Jedna od trigonometrijskih funkcija. Većina računara poseduje [illegible] druge funkcije kao što su COS, TAN, ATN, SQR, EXP, LOG i slično.

USR (n)



Počinje izvršavanje mašinskog programa koji se nalazi na apsolutnoj adresi n. Ukoliko ne poznajete mašinsko programiranje A[illegible]USR ([illegible] će verovatno izazvati bl[illegible]je računara [illegible] u tom slučaju moraćete da ga isključite i ponovo uključite.

```
10 PRINT "KOLIKO IMAS GODINA"
20 INPUT A
30 IF A>16 THEN PRINT "PRESTAR"
40 IF A<16 THEN PRINT "PREMLAD"
50 IF A=16 THEN PRINT "KOLIKO TREBA!"
60 END
RUN
KOLIKO IMAS GODINA
?20
PRESTAR
```

- = jednakost
- <> različito, naredba iza THEN se izvršava samo ako su brojevi (reči) različiti
- > veće, naredba iza THEN se izvršava samo ako je prvi broj veći od drugog, neki računari omogućavaju da se na ovakav način poredа i reči i tada je „manja" reč ona koja je pra po abecednom redu,
- < manja, oblik suprotan prethodnom
- <= manja ili jednako, naredba iza THEN se izvršava samo ako je prvi broj manji od drugog ili su brojevi jednaki,
- >= veće ili jednako, naredba iza THEN se izvršava samo ako je prvi broj veći od drugog ili su oni jednaki

U primeru koji je dat na slici računar najpra pita korisnika za starost, a zatim daje komentar njegovog odgovora

Naredba GO TO

Ovu naredbu programeri-početnici najlakše shvataju. Iza nje se mora nalaziti linijski broj neke naredbe. U tom slučaju naredba GO TO radi baš ono što joj ime i kaže (GO TO=idi na) — nastavlja izvršavanja programa od naredbe čiji je linijski broj naveden. Naredba GO TO 100 izvršava program od linije čiji je broj 100 (ukoliko ta linija ne postoji neki računari prijavljuju grešku a drugi nastavljaju izvršavanje programa od prve linije čiji je broj veći od 100)

Naredba GO TO je prva „opasna" naredba koju smo upoznali. Zamislite da se u vašem programu nehotice nađe naredba 100 GO TO 100. Računar će neprekidno izvršavati tu naredbu i vraćati se na nju. Vi ni na koji način nećete biti obavešteni da računar obavlja ovaj Sizifov posao osim što ćete primetiti da se vaš program predugo izvršava. Šta da radite u tom slučaju? Možete, jasno, da isključite računar iz struje, ali će time i čitav program biti obrisan. No, negde na tastaturi se nalazi i taster koji je obično obeležen sa BREAK (BREAK=slomi, u ovom slučaju prekini). Pritisak na ovaj taster izaziva prekid izvršavanja programa. Računar obično ispisuje koju je programsku liniju izvršavao kada ste mu naredili da prekine sa radom, pa možete lako da proverite šta sa njom nije bilo u redu

```
10 INPUT K$
```

Ove dve linije nalažu kompjuteru da pređe na drugi deo programa

```
10 INPUT K$
20 IF K$="DA" THEN GO TO 100
30 IF K$="NE" THEN GO TO 200

100 PRINT "OTKUCALI STE DA"
110 GO TO 210

200 PRINT "OTKUCALI STE NE"
210 END
```

Matematički program

```
10 PRINT "OTKUCAJ BROJ"
20 INPUT A
30 PRINT "OTKUCAJ JOŠ JEDAN BROJ"
40 INPUT B
50 PRINT "ZELITE LI DA
60 PRINT "SABIRAM, ODUZIMAM, MNOŽIM"
70 PRINT "DELIM ILI PREKINEM"
80 INPUT C$
90 IF C$="SABERI" THEN PRINT A+B
100 IF C$="ODUZMI" THEN PRINT A−B
110 IF C$="MNOZI" THEN PRINT A−B
120 IF C$= "DELI" THEN PRINT A/B
130 IF C<>"PREKINI" THEN GO TO 10
140 END
```

Program će prekinuti sa radom tek kada otkucate reč PREKINI

```
RUN
OTKUCAJ BROJ
?17
OTKUCAJ JOŠ JEDAN BROJ
?184
ZELITE LI DA
SABIRAM, ODUZIMAM, MNOZIM,
DELIM ILI PREKINEM
?SABERI
201          ← rezultat
OTKUCAJ BROJ
?
```

```
    A•B
130 IF C$<>
```

Naredba GO TO može da se nađe i iza naredbe THEN. Ako je uslov koji je ispitivan u okviru naredbe IF ispunjen, izvršavanja programa se nastavlja od programske linije čiji je broj naveden iza GO TO. Ukoliko uslov nije ispunjen, naredba iza THEN (u ovom slučaju GO TO) se jednostavno preskače i računar nastavlja izvršavanje programa od naredbe koja se nalazi iza IF.

Ovaj jednostavni program omogućava korisniku da ispita računarevo znanje matematike. Računar najpra pita korisnika za dva broja a zatim ga pita šta treba da se uradi sa ovim brojevima. Korisnik kuca reč koja označava traženu računsku operaciju (može da se unese SABERI, ODUZMI, POMNOŽI, PODELI ili PREKINI). Računar zatim ispituje koju je reč korisnik otkucao, izvršava traženu računsku operaciju i prikazuje rezultat na ekranu. Posle prikazivanja rezultata računar nailazi na naredbu GO TO 10 i vraća se na sam početak očekujući da mu korisnik postavi novi problem. Prekid rada ovog programa je obezbeđen — korisnik samo treba da otkuca bilo koja dva broja i da onda, kada računar pita šta treba da se radi sa brojevima, otkuca PREKINI. Računar poslušno prekida sa radom. Drugi način da se prekine izvršavanje programa je već opisani taster BREAK

Ciklusi i procedure

U programima je često potrebno da se jedna sekvenca ponovi nekoliko puta. (Čitavo ovo razmatranje je potrebno da bismo omogućili da računar određuje pobednika u igri „Svemirska potera". Računar naime treba da 60 puta ponovi pitanja o koordinatama protivnika i da, ako igra do tog momenta nije završena, proglasi prvog igrača za pobednika) U programiranju se deo programa koji se ponavlja određen broj puta naziva ciklusom. Ciklus je, dakle, grupa instrukcija u programu koja se izvršava određeni broj puta posle čega se izvršavanje programa nastavlja instrukcijama koje slede posle ciklusa. Nasuprot ciklusu odomaćen je termin procedure za deo programa koji se izvršava beskonačno mnogo puta. Reč „beskonačno", naravno, treba shvatiti uslovno: procedura se izvršava sve dok korisnik, da bi je zaustavio, ne pritisne spasonosni taster BREAK. Kod programera-početnika ciklusi se lako pretvaraju u procedura — dovoljna je mala greška i računar neće imati dobro uputstvo o izlasku iz ciklusa, pa će se njegov rad pretvoriti u beskonačnu proceduru. Ne treba, međutim, misliti da su procedura uvek posledice greški programera. Program koji pretvara računar u štopericu predstavlja primer korisno upotrebljene procedura: računar neprekidno pokazuje tekuće vreme sve dok ga korisnik ne zaustavi

*Osnovna karakteristika ciklusa je, dakle, preciziranje nekog uslova, posle čijeg ispunjavanja računar završava ciklus. Ovaj uslov (u programiranju ga nazivaju **izlazni kriterijum**) ilustrovan je u prethodnom primeru — računar prikazuje rezultate sabiranja, oduzimanja, množenja i deljenja brojeva koje korisnik kuca sve dok ovaj ne otkuca **PREKINI** posle čega računar završava sa radom*

Najčešće je potrebno da se neki deo programa ponovi unapred određeni broj puta. Za to služe naredbe FOR i NEXT. Naredba FOR zahteva da bude napisana na određeni način koji će postati jasan iz datog primera. Računar najpra 8 puta ispisuje 2 PLUS 2 JE 5, a zatim dodaje — SAMO SAM SE SALIO. Kako je to postignuto? Naredba FOR J=1 TO 8 (u prevodu „od J jednako 1 do 8) nalaže računaru da izvrši sledeće naredbe za vrednosti promenljive J 1, 2, 3, 4, 5, 6, 7, i 8. Sledeće naredbe nam je dobro poznata — PRINT prenosi tekst između navodnika na ekran. Sledi naredba je NEXT J (NEXT J= sledeće J). Ona označava kraj ciklusa i ima višestruko dejstvo: najpra se sadržaj promenljive J povećava za jedan, a onda računar ispituje da li je J postalo veće od 6 (setimo se da je naredba FOR glasila FOR J=1 TO 8). Ukoliko je odgovor na ovo pitanje odrečan, izvršava se ponovo čitav ciklus. Ali, ukoliko je J postalo veće od 8 ciklus se

Pogrešno sabiranje

Ciklus

```
10 FOR J=1 TO 8
20 PRINT "2 PLUS 2 JE 5"
30 NEXT J
40 PRINT
50 PRINT "SAMO SAM SE SALIO!"
60 END
```

Neki kompjuteri nemaju znak uzvika pa ćete morati da ga izostavite

```
2 PLUS 2 JE 5
2 PLUS 2 JE 5
2 PLUS 2 JE 5
2 PLUS 2 JE 5
2 PLUS 2 JE 5
2 PLUS 2 JE 5
2 PLUS 2 JE 5
2 PLUS 2 JE 5
SAMO SAM SE SALIO
```

```
10 CLS
20 FOR J=7 TO 1 STEP −1 prvi ciklus
30 PRINT ,,OSTALO JE JOS ",J," DANA RASPUSTA"
40 NEXT J
50 PRINT ,,EKSPLODIRAĆU!"
60 FOR J=1 TO 1000                 ciklus koji služi samo da odloži
70 NEXT J                          ,,eksploziju"
60 PRINT ,,BOOOOOM!'
90 END
```

završava i računar nastavlje od naredbe 50 koje nalaže prikazivanja komentare i prestanak reda

Naredba FOR ima još jedan oblik FOR TO STEP Iza STEP mora da se nalazi broj (ili ime promenljiva) koja označave za koliko će se pri svakom nailasku ne naredbu NEXT, menjali sadržaj kontrolne promenljive. De smo u prethodnom primeru upotrebili naredbu 10 FOR J=1 TO 8 STEP 2, ciklus bi se izvršavao za J=1 3 5 i 7 što znači da bi sa na akranu samo 4 puta pojevio tekst ,2 PLUS 2 JE 5" Ovej korak (STEP=korak) može da bude i negativan, kao što se vidi u sledećem primeru U programu se nalaze dva cikluse Računar najpre prebrojeva koliko mu je dane školskog raspusta preostalo (7 6 1) i javlja de će ,eksplodireti" kade mu vrame istekne Zatim stupa na scenu sledaći program računar na ekranu neprekidno ispisuje tekuće vreme od momenta kada je program startovan Naravno, časovnik će u početku raditi mnogo brže nego što je potrebno De biste ga doterell, morete da dodete linije 50 FOR I=1 TO 100 i 60 NEXT I koja će produžiti vrema izvršavanja svakog ciklusa Broj 100 je stavljen primera radi — morete da procenite brzinu reda vašeg računare i da podesite broj prema toj brzini Za ovo će vem biti potrebno dosta strpljenja i — preva štoperica

Na sledećoj slici je prikazan primar najčešće greška koju početnici preva Rekli smo da jedan (ili čak nekoliko) ciklusa može de se nelazi unuter drugog Međutim ciklusi ne smaju da se ,,seku", kao što je to slučaj kod datog programe Ukoliko napravite ovekav program, rezultati su nepredvidljivi (zavise od računara i nekih drugih paremetere), ali je jedno sigurno — program neće funkcionisati oneko kako ste želeli.

Greške kod ciklusa

```
10 FOR I=1 TO 4
20 FOR J=1 TO 4
30 PRINT I
40 PRINT J
50 NEXT I
60 NEXT J
```



Ciklusi ne smaju da sa seku — jedan mora da buda unutar drugog

```
JOS SAMO 7 DANA RASPUSTA
JOS SAMO 6 DANA RASPUSTA
JOS SAMO 5 DANA RASPUSTA
JOS SAMO 4 DANA RASPUSTA
JOS SAMO 3 DANA RASPUSTA
JOS SAMO 2 DANA RASPUSTA
JOS SAMO 1 DANA RASPUSTA

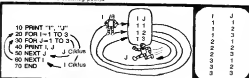
EKSPLODIRACU!

BOOOOOM
```

Neki računari su sporiji od drugih pa ćeta morati da stavite manji broj (npr. 500 ili 250) na liniju 60



ciklus koji ništa i ne radi — njegov cilj je da uspori rad programa i da korisniku na znenja šta će se dogoditi Zatim naredba PRINT ,BOOOOM" simulira eksploziju Računaru jasno neće ništa biti čak ni posla oveko žestokog pucnja



I	J
1	1
1	2
1	3
2	1
2	2
2	3
3	1
3	2
3	3

U prethodnom primeru videli smo da dve ciklusa mogu da se nađu jedan iza drugog Nema međutim, nikakve prepreke da se jedan ciklus nađe unutar drugog U sledećem primaru vidimo da se ciklus koji kontroliše promenljiva I izvršava tri pute i da se, u toku svakog izvršavanja ciklusa I, izvrši po tri puta ciklus koji kontroliše promenljiva J Na ekranu vidimo kako su se menjali sadržaji promanljivih u toku izvršavanje ciklusa

Već smo pomenuli upotrebu računare kao digitalnog časovnika To nam omogućava sledeći

Da li ste čuli da . . .

je prvi programer u istoriji, u stvari, programarka! To ja bila Augusta Ada, Lady Lovelace inače jedino date lorda i ledi Bajron (Byron) Ona je 1843 u časopisu Scientific Memories objavila ozbiljan članak o programiranju Bebidžove (Charles Babbage) računske mašine i sličnih automata Ladi Lovelace, inače i sama matematičar, pretpostevlja da bi bilo moguće napraviti mašinu koja bi od čoveka primala naloga na standardnom englaskom jeziku i izvršavala ih neobično brzo Ladi Lovelace u pomenutom radu izlaže i dva ograničenja mašina će moći da rešava samo probleme za koje se precizno može definisati postupak rešavanja i neće moći da ,,razume" sve finese jezičkih konstrukcija — beće neophodno da se sastavi neki jednoznačni ,,podjezik" koga ona naziva ,,programski jezik" (termin ja i danas u upotrebi) Fascinantno!

```
10 CLS
20 FOR M=O TO 59
30 FOR S=O TO 59
40 PRINT M," ",S
70 CLS
60 NEXT S
90 NEXT M
100 END
```

0:45

Da "naštalujete sat" dodajte naredbe 50 FOR Z=1 TO 100 60 NEXT Z



Svemirska potera 2

Sada znamo dovoljno da napravimo pravi program za igru „Svemirske potera" Program će jasno biti u mnogoma sličan prathodnom ali će ovoga puta računar biti u stanju da odredi koji je igrač pobedio Ukoliko ste pažljivo čitali dosadašnji tekst verovatno ćete moći i sami da sastavite ovaj program Ukoliko vam to ne uspe, pročitajte našu verziju Možda će vam njenom analizom postati jasne neke pojedinosti koje do sada niste razumeli

```
 10 FOR K=1 TO 60
 20 PRINT ,,X KOORDINATA ULJEZA?"
 30 INPUT A
 40 PRINT ,,Y KOORDINATA ULJEZA?"
 50 INPUT B
 60 CLS
 70 PRINT ,,X KOORDINATA BRANIOCA?"
 80 INPUT C
 90 PRINT ,,Y KOORDINATA BRANIOCA?"
100 INPUT D
110 CLS
120 LET X=SQR((A-C)*(A-C)+(B-D)*(B-D))
130 PRINT ,,TRENUTNO RASTOJANJE JE",
140 PRINT X, ,,ZVEZDANIH JEDINICA"
150 IF X<1.5 THEN GO TO 230
160 PRINT ,,PRIPREMITE SE ZA DAVANJE"
170 PRINT ,,NOVIH POZICIJA"
180 NEXT K
190 PRINT ,,BRANILAC NIJE USPEO DA"
200 PRINT ,,UNISTI ULJEZA! ULJEZ JE"
210 PRINT ,,POBEDIO!!!!"
220 GO TO 260
230 PRINT ,BRANILAC JE STIGAO !"
240 PRINT  UNISTIO ULJEZA!"
250 PRINT  BRANILAC JE POBEDIO!!!"
260 END
```

ukoliko je rastojanja manja od 1,5 jedinica, pralazi se na naredbu broj 230

ovo je kraj ciklusa koji je započeo na liniji 10. Posle 60 poteza smatra se da je prvi igrač dobio partiju i računar automatski izvršava naredbu broj 190

Ovaj kratki strip (uz prateći program) iluatruje potrabu za potprogramima Zamisiimo da trabe robotu da damo uputstvo da ode u samoposlugu i kupi neku stvar Uputstvo za odlazak u samoposlugu je relativno složeno, zašto da ga ponavljamo svaki put kada nam je potrabna neka sitnica? Zar ne bi bilo jednostavnije jednom za svagda naučiti robota kako se ide u samoposlugu, a docnije mu samo govoriti šta treba da kupi? To je kao što se iz programa vidi, omogućiia upotrebe potprograma







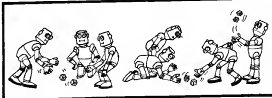
Potrabu za potprogramima ćemo ilustrovati na još jednom primaru Ovoga puta program će biti nešto složeniji, pa ćemo utrošiti nešto više prostora da ga objasnimo Igra koju program omogućava se kod Amerikanaca, naziva „Jive Turkey" (brbljiva ćurka) i predstavlje jednu od retkih igara u kojoj traba kompjuter naučiti da „laže" Na početku igra kompjuter „zamišlja" broj između 0 i 999 i izaziva igrača da pogodi taj broj igrač pokušava da pogodi broj, a računer ga obaveštave da li je broj koji je igrač otkucao veći ili manji od „zamišljenog" broja No, tu računar nija previše iskren — u 80% slučajeva on će dati tačnu informaciju, ali će u preostalih 20% slučajeva namarno dati pogrešan podatak

Prva stvar koju trabe da rasčistimo je kako računar može da „zamisli" broj Većine računara ima mogućnost da generiše slučajne brojeve koji se najčešće korista u igrama „Slučajan" broj nije slučajan u punom smislu reči — on je izračunat nekim složenim postupkom na osnovu sadržaja nekih registara računara koji se neprekidno menjaju ali obzirom da korienik ne zna koji je postupek primenjan, za njega je ovaj broj slučajan Slučajni brojevi se bezmalo kod svih računara, generišu pomoću naredbe RND Naredbe RND generiše alučajan broj između 0 i 1, a ako je korisniku potraban ceo slučajan broj između D i A, on koristi naredbu INT (RND*A) Ovde je INT jedna od specijalnih funkcija koja vrši izračunavanja celog dela broja koji se nalazi u zagradi

Postoje računari koji poseduju i nešto složeniji oblik naredbe RND Kod ovih računara može direktno da se generiše ceo broj između O i A naredbom RND (A) Kod ovih računara se za

Potprogrami

...program je neka vrsta mini-programa ... predstavija deo glavnog programa On ...avi a neki konkratan zadatak koji traba da se ...zvrši veći broj puta Kada je glavnom ...gramu neophodno da se taj zadatak obavi, ...ednostavno „poziva potprogram" i po ...om zvršavanju nastavija sa redom ...grami su neobično korisni — ne samo ...program postaje kraći i pregledniji ...se štedi na memoriji i omogućava ...strukturalno programiranja



```
 10 PRINT  ŠTA TREBA DA KUPIM U
RADNJI"
 20 INPUT X$
 30 GO SUB 100
 40 PRINT  JOŠ NEŠTO?"
 50 INPUT M$
 60 IF M$=  NE  THEN GO TO 1000
 70 GO TO 10
100 PRINT  PRODJI KROZ VRATA  SKRENI"
110 PRINT "LEVO, PONOVO SKRENI LEVO"
120 PRINT "UDJI U RADNJU  KUPI ",X$
130 PRINT 'VRATI SE KUCI"
140 RETURN
1000 END
```

ova naredba nalaže računaru da pređe na izvršavanja naredbe 100

ova naredba nalaža računaru da se vrati u glavni program, u ovom slučaju na naredbu broj 40

naredba 1000 je END i izaziva prekid rada

```
PRINT RND
662741814
PRINT RND(99)
77
```



Ili RND(O), ili RND(1) kod nekih računara

genarisanje decimalnog broja između 0 i 1 konsti naredba RND(0) ili RND (1). Mi ćemo pretpostaviti da posedujemo računar koji ime samo najjednostavniji oblik naredbe RND — onu koja generiše brojeve između 0 i 1

```
10 CLS
20 PRINT „ZAMISLICU   BROJ  IZMEĐU   1 I 999"
30 GOSUB 1000                       prvi poziv potprograma — „zamišljanje"
40 LET X=N                          broja
50 PRINT „POGAĐAJ BROJ"
60 INPUT P                          računar ponovo predlaže igraču da
70 IF X=P THEN GO TO 160            pogađa broj.
80 GOSUB 1000
90 IF X>P THEN GO TO 130
100 IF N>800 THEN GO TO 140         drugi poziv potprograma — računar
110 PRINT „BROJ JE MANJI OD ";P     odlučuje da li će dati istinit odgovor.
120 GO TO 50
130 IF N>800 THEN GO TO 110
140 PRINT „BROJ JE VEĆI OD ";P
150 GO TO 50
160 PRINT „BRAVO!"                  broj je pogođen — automatski počinje
170 PRINT „ZAMISLIĆU NOVI BROJ"     nova partija
180 GO TO 30
1000 LET N=INT(RND*1000)            potprogram za genereisanje slučajnih
1010 RETURN                         brojeva između 0 i 999.
```

Računar najpre briše sadržaj ekrana i štampa zaglavlje (pravila igre). Zatim nailazi na naredbu GO SUB 1000 koja kao što smo videli, izaziva prelazak na probramsku liniju 1000 koju računar shvata kao početak potprograma. Tamo je generisan slučajan broj između 0 i 999 i smešten u memoriju računara pod imenom N. Zatim računar nailazi na naredbu RETURN koja izaziva pogratak u glavni program. Potom prenosi sadržaj promenljiva N u promenljivu X. Znači broj koji je računar „zamislio" se zove X

Sada računar traži od igrača da pogađa broj. Igrač kuca neki broj između 0 i 999 koji računar naziva P. Ukoliko je P=X, naredba 70 izaziva prelazak na naredbu 160 gde se igraču odaje zasluženo priznanje na pogotku

Linija 80 je, na prvi pogled, nepotrebna. Računar generiše još jedan slučajan broj. Zašto? Računar treba da odredi da li treba igraču da pruži tačan ili lažan odgovor. Da bi to odlučio on generiše još jedan slučajan broj između 0 i 999 i ispituje da li je taj broj veći ili manji od 800. U otprilike 60% slučajeva računar će dobiti slučajan broj manji od 800 i zato pružiti igraču tačan odgovor. U otprilika 20% slučajeva ovaj broj će biti veći od 800 pa će računar „lagati" igrača

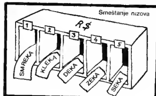
Dalji tok programa ilustruje naredbe IF THEN i trebao bi da bude jasan na osnovu komentara koji su dati uz njega

U ovom primeru potprogram je bio potreban obzirom da računar dve puta treba da izabere broj između 0 i 1000. Mogla su se, jasno, koristiti dva RND naredbe ali program na ovaj način može lakše da se razume a zauzima i manje memorije. Ušteda bi bila još veća da je potprogram duži i da se poziva više puta

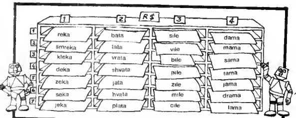
Rad sa nizovima

*Do sada se svaki broj koji smo unosili u memoriju računara nazivao nekim slovom između A i Z. Vrlo često je, međutim, potrebno da se u memoriji računara formira niz podataka koji su po nečemu srodni, pri čemu se svaki od njih može posebno čitati i (po potrebi) menjati. Zamislimo, na primer, da želimo da napišemo program koji će pisati dečje pesmica koje će, svaki put kada startujemo program, biti u nečemu različite. Odlučili smo da se na kraju prvog stiha nalazi reč REKA. Sledeći posao je da zamislimo nekoliko reči koje se rimuju sa reka (npr smreka, kleka, deka, zeka i seka) i da naučimo računar da jednu od njih slučajno izabere i smesti je na kraj drugog stiha. Sve ove reči imaju jednu zajedničku osobinu (rimuju se sa reka) a ipak svaka mora da se koristi posebno. Zato ćemo formirati niz R$ koji će imati pet elemenata. Formiranje ovoga niza moramo na početku programa da najavimo naredbom DIM R$ (5) (DIM je skraćenica od DIMENSION, ovom naredbom se daje dimenzija niza odnosno saopštava računaru koliko elemenata niz sadrži). Sledećih nekoliko naredbi u programu će biti LET R$ (1)= „SMREKA" LET R$ (2)= „KLEKA", LET R$ (3)=„DEKA" LET R$ (4)= „ZEKA" i LET R$ (5)=„SEKA". Kada, docnije, bude bilo potrebno da se slučajno izabere jedna od ovih reči, generisaćemo slučajan broj između 1 i 5 (naredba za to je LET A=INT (RND*5+1)) i štampati reč R$ (A). Vašim poetskim i programskim sposobnostima ostavljamo da sastavite čitav program za pisanje pesmica*

Umesto da u nizova smeštamo reči, možemo da smeštamo brojeve. Na ovaj način se obično smeštaju koordinata nekog višedimezionalnog vektora, pa se zato nizovi podataka koje smo upoznali ponekad nazivaju i „vektori" ili, češće „jednodimenzionalni nizovi". Za razliku od njih postoje i dvodimanzionalni nizovi ili, kako se češće nazivaju, „matrica". Matricu možemo da

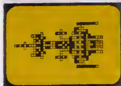


zamislimo kao niz nekoliko nizova. ako nam posmatrajući prethodni primer „poraste apetit" pa poželimo da nam ni poslednje reč prvog stiha ne bude stalno iste, moraćemo da oformimo tabelu (matricu) grupe reči koje se rimuju. Jedna takva matrica je data na sledećoj slici. Ovakvu matricu treba dimanzionisati primenom naredbe DIM R$ (4 7) a zatim je, primenom naredbi LET popuniti potrebnim rečima. Zatim ćemo generisati slučajan broj između 1 i 4 (nazovimo taj broj I) i drugi slučajan broj između 2 i 6 (nazovimo taj broj J), pa ćemo prvi stih završiti rečju R$ (I 1) a drugi — rečju R$ (I J)

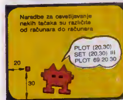


Crtanje pomoću kompjutera

Crtanje po ekranu je jedna od izuzetno zanimljivih disciplina, pogotovu za mlađe korisnike računara. Naredbe koje omogućavaju crtanje su toliko različite od računara do računara da možemo da damo samo njihove najosnovnije karakteristike



Ekran je podeljen na određeni broj tačaka (u stranoj literaturi svaka tačka se naziva „pixel"). Računar sa velikom memorijom i visokom rezolucijom može da nacrta slika koja se sastoje od desetina hiljada tačkica u različitim bojama dok jeftiniji računari mogu na ekranu da prikažu slike koje se sastoje od samo par stotina tačaka. U vašem uputstvu za upotrebu računara piše koliko tačaka na ekranu možete da kako to programeri kažu, adresirate, kao i gde se nalazi referantna tačka. Ova referentna tačka se obično nalazi u gornjem levom ili donjem levom uglu ekrana i kordinate svih tačaka se računaju prema njoj. U okviru uputstva za upotrebu je data i mapa ekrana koje govori o tome gde se, tačno, nalazi koja tačka. Da biste osvetlili bilo koju tačku, koristite naredbu SET X Y (kod nekih računara se korijeti naredba PLOT X,Y ili neka druga) gde su X i Y koordinate tačke koju treba osvetliti



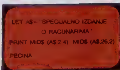
Suprotna naredba (RESET X,Y odnosno UNPLOT X,Y) se koristi za „gašenja" neke tačke koja je ranije osvetljena. Kod računara koji rade sa televizorom u boji, potrebno je dati i komandu kojom bojom treba osvetliti neku tačku

Jednostavni demonstracioni program koji je dat čeka da korisnik otkuca dva broja (koordinate neke tačke), a zatim osvetljava tu tačku. Program je vrlo koristan za one koji žele da steknu osećaj o tome gde se koja tačka nalazi. Program će neprakidno očekivati od korisnika da kuca koordinate tačaka sve dok ovaj ne pritisne BREAK — program dakle, predstavlja primer korisno upotrebljene procedura

Rad sa rečima

Do sada smo upoznali način da alfanumeričkim promenljivima dodeljujemo vrednosti i da ove posle prikazujemo na ekranu. U nekim slučajevima će se pokazati neophodnim da menjamo ili analiziramo sadržaje nekih numeričkih promenljivih. Način na koji se ovo postiže u mnogome zavisi od računara. Većina kompjutera ima posebne funkcije LEFT$, RIGHT$ i MID$. Za korisnike je najpogodnije da se priviknu na korišćenje ove poslednje funkcije pošto su prve dve njeni specijalni slučajevi. Funkciju MID$ ćemo objasniti na jednom jednostavnom primeru

Sastavimo program koji na početku ima naredbu
10 LET A$= SPECIJALNO IZDANJE O RAČUNARIMA". Ukoliko dodamo naredbu 20 PRINT MID$ (A$ 12,7) i izvršimo program sa RUN, na ekranu će se pojaviti reč IZDANJE. Računar je izvršavajući naredbu MID$, izdvojio 7 slova koja se nalaze u reči A$ i to počevši od 12 slova. Da smo upotrebili 20 print MID (A$,2,4), MID$ (A$,26,2) na ekranu bi se pojavila reč PECINA. Pokušajte sami da zaključite zašto!



ZX računari i reči

Sve do sada pomenute naredbe mogle su da se koriste na ZX 81 i Spectrumu. Kod reda sa rečima, međutim, ova kompatibilnost prestaje. Rad sa rečima je ovde organizovan na drukčiji način, teško je reći da li je ovaj način bolji ili lošiji od standardnog, ali nema sumnje da pruža iste mogućnosti

Kod ZX računara svaku reč zamišljamo kao niz slova. Ako se na primer, na početku programa nalazila naredba DIM A$ (10) računar će oformiti jednu promenljivu koja će se zvati A$ i predstavljati tekst od 10 slova (kod drugih računara ova naredba bi oformila niz A$ koji bi se sastojao od 10 reči). U svaki element ovog niza (niz, sumarno, predstavlja promenljivu) smešta se po jedno slovo. Ako na primer ZX81 izvrši naredbe
10 DIM A$ (40)
20 LET A$=„SPECIJALNO IZDANJE O RAČUNARIMA
30 PRINT A$ (1)
na ekranu će se pojaviti slovo S. Razlog za to je što je, po izvršenju naredbe 20 A$ (1) postalo S, A$ (2) — P i slično (A$ (31) je postalo A). U daljem toku programa mogu ravnopravno da se koriste promenljiva A$ (koja predstavlja čitav tekst) i neki njeni delovi

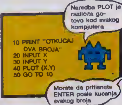
Da iz reči A$ izdvojimo pojedinu grupu slova, koristimo naredbu TO. Da, na primer, izdvojimo reč „IZDANJE" koristićemo 40 PRINT A$ (12 TO 18) a da izdvojimo reč PECINA.
50 PRINT A$ (2 TO 5), A$ (26 TO 27)

Ukoliko ne navedemo broj ispred TO, računar smatra da treba da počne od početka reči (PRINT A$ (TO 4) bi izdvojilo reč SPEC) a ako ne navedemo broj iza TO — do kraja. Na ovaj način naredbe LEFT$, RIGHT$ i MID$ su postale nepotrebne

Često se ukazuje potreba za radom sa nizovima i matricama reči. Kod ZX računara sve ostaje kao i do sada samo se, u DIM naredbama, dodaje jedna dimenzija više — ova dimenzija označava maksimalni broj slova koji svaka alfanumerička promenljiva može da sadrži. Naredba 10 DIM B$ (10,20), na primer, nalaže računaru da u memoriji rezerviše prostor za deset alfanumeričkih promenljivih (zvaće se B$ (1), B$ (2), ... B$ (10)) od kojih će svaka moći da primi najviše 20 slova. Kod matrica se dodaje još jedna dimenzija koja označava broj redova.

Kraj škole

Ovo je ujedno i kraj „Osnovne škole bejzika". Ako ste je pažljivo pročitali, naučili ste mnoštvo bejzik naredbi koje su vam sasvim dovoljne čak i za pisanje najsloženijih programa. Ostalo je jasno, mnogo toga što na ovim stranicama nije ni pomenuto. Razlog za to je što su računari različiti i imaju različite mogućnosti, dok smo se mi odlučili da damo samo one naredbe koje su zajedničke za sve. I pored toga, neke od spomenutih naredbi mogu da budu drukčije na vašem računaru (najlakše bi trebalo da bude onima koji poseduju ZX računare i kompjutere sa standardnim microsoftovim bejzikom), pa ćete morati da konsultujete vaše uputstvo za upotrebu računara da biste ih primenili. Ova škola vam, dalje, ne može zameniti uputstvo za upotrebu (predlažemo vam da ga još jednom pročitate — možda ćete sada usvojiti nešto što ranije niste primetili ili razumeli) ali je pokušala da vas, na popularan način i uz mnoštvo slika, uvede u osnovne tajne bejzika i omogući vam da samostalno napredujete.



RUN
OTKUCAJ DVA BROJA
?24
?24
prva tačka
OTKUCAJ DVA BROJA
?30
?15
druga tačka

S KNJIGOM ĆE SVE BITI LAKŠE

PRIRUČNICI, PERIODIKA, KASETE, ZA SVE ZALJUBLJENIKE U RAČUNARE, IGRAČKE NAŠEG DOBA — KOJI SU I VIŠE OD TOGA

Najveća slovenačka knjižara — Mladinska knjiga u Ljubljani, Titova br. 3 — nudi vam, na svom posebnom računarskom odeljenju, najbogatiji i najsvestraniji izbor stručne i popularne literature s područja računarstva i informatike:

Literatura na slovenačkom jeziku:

1.	Gerlič: ABC računalništva	400 din
2.	Špiler, BASIC	1000 din
3.	Ellershaw, Schofield: BASIC za začetnike	400 din
4.	Mitrović, Ivanović, Balić: OSEBNI računalnik	550 din
5.	Laurie: Čudoviti svet računalnikov	2600 din
6.	Uvod v računalništvo, Prvi koraki v BASIC-u, Učenja z računalnikom, Grafične in zvočne igre — sve 4 knjige	3600 din
7.	Bratković: Metode programiranja	714 din
8.	Bratković: Programiranja računalnikov	965 din
9.	Bratko: Inteligentni informacijski sistemi	680 din
10.	Divjak: Računalniška grafika	476 din
11.	Matko: Diskretni regulacijski sistemi	1100 din
12.	Furlan: Osnove nelinearnih elementov	510 din
13.	Kodek: Poglavja iz računalniških sistemov	280 din
14.	Kodek: Uvod v mikroprocesorske sisteme	680 din
15.	Ogorelec: Nove smeri v zaščiti in lokalni avtomatizacij	680 din
16.	Virant: Zanesljivost računalniških sistemov	425 din
17.	Kozak: Podatkovne strukture in algoritmi	1750 din
18.	Wirth: Računalniško programiranje 1. del	1000 din
19.	Wirth: Računalniško programiranja 2. del	600 din
20.	Mohar, Zakrajšek: Uvod v programiranje	400 din
21.	Matko: Metode optimiranja I	600 din
22.	Zakrajšek: FORTRAN	600 din
23.	Programski jezik FORTRAN (Iskra-Delta)	480 din
24.	Hišni računalnik od A do Ž (u pretplati do izlaska iz štampe)	2800 din

Literatura na srpskohrvatskom jeziku:

25.	Reljić, Makanec: BASIC	900 din
26.	Parezanović: Programski jezik BASIC	532 din
27.	Stojković, Tošić: BASIC, zbirka zadataka iz progr.	600 din
28.	Marković: ASEMBLER simbolički progr. jezik	1076 din
29.	Stojković, Tošić, Stojmenović: Progr. jezik PASCAL	700 din
30.	Lovrić: PASCAL	700 din
31.	Turajlić: Osnove programiranja i sistema Fortran IV	650 din
32.	Janković: Programski jezik COBOL	350 din
33.	Tkalec: Programski jezik COBOL	760 din
34.	Nedrah: COBOL	900 din
35.	Parezanović: Mašinski i simbolički PDP jezik	504 din
36.	Salčić: Mikroračunarski sistemi	680 din
37.	Grundler: Uvod u mikroprocesore	600 din
38.	Ribarić: Arhitektura mikroprocesora	1000 din
39.	Sobotka: Mikroprocesori i mikroračunala	1450 din
40.	Đurić: Mini i mikro računari	1200 din
41.	Kvaternik: Uvod u operativne sisteme	600 din
42.	Hansen: Principi operativnih sistema	450 din
43.	Petrić: Nelinearno programiranja	1000 din
44.	Vučković: Linearno programiranja	790 din
45.	Stajić: Sredstva za obradu podataka	525 din
46.	Erbes: Uvod u računarske mreže	350 din
47.	Altman: Osnovi teorije diskretnog modeliranja	475 din
48.	Krčevinac: Algoritmi i programi iz operac. istraživanja	1200 din
49.	Petrić: Operaciona istraživanja — zbirka zadataka 1	808 din
50.	Petrić: Operaciona istraživanja — zbirka zadataka 2	730 din
51.	Turk, Bodin: Računarska grafika	490 din
52.	Turk, Bodin: Analiza primjenom računala	665 din
53.	Nožica: Analiza primjenom računala — primeri i zadaci	1196 din
54.	Žiljak: Simulacija računalom	600 din
55.	Smiljanić: Osnove digitalnih računala	900 din
56.	Smiljanić: Mikroračunala	865 din
57.	Malvino: Osnovi i primjena digitalne tehnike	700 din
58.	Zupan: Uvod u komutacione sustave	420 din
59.	Katanić: Uvod u programiranje	1100 din
60.	Više autora: Programiranja NC i CNC mašina alatki	620 din
61.	Aleksić: Računari — organizacija i arhitektura	610 din
62.	Belak: Osnove poslovnih informacijskih sistemov	820 din
63.	Birola: Kompjuterska obrada podataka	400 din
64.	Mrakovčić: Standardizacija programiranja u ERC	600 din
65.	Ferišak: Organizacija EOP	600 din
66.	Panian: Upravljanje djelotvornošću EOP	450 din
67.	Matasović: Primjena elektr. računara u izvedbi investicionih projekata	650 din
68.	Šimić: Osnovi kibernetike	340 din
69.	Marjanović: Primjena kibernetike u rukovod. rad. organ.	550 din
70.	Aranđelović: Kompjuterska organizac. u savremenoj privredi	1000 din
71.	Turajlić: Sistem automat. upravlj. — zbirka zadataka	990 din
72.	Matković: Teorija informacije	1400 din
73.	Srića: Sistem — informacija — kompjuter	760 din
74.	Ferišak: Osnove informatike	1200 din
75.	Han, Balaban: Osnovi informatike	650 din
76.	Više autora: Informacijski sistemi	550 din
77.	Stanković, Tomović: Nelinearni sistemi autom. upravljanja	840 din
78.	Popović: Elektronski računar — most u budućnost	900 din
79.	Više autora: Lični kompjuter	380 din
80.	Evans: Kompjuterski izazov	600 din
81.	Čavić: Engleski jezik u informatici	665 din
82.	Čavić: Sintakt. osobenosti engl. jezika u informatici	320 din
83.	Više autora: Obrada podataka i programiranja — petojezični rečnik	3200 din
84.	Matić: Principi upravljanja komutac. sistemima	860 din
85.	Simić: Osnovni automatskog upravljanja	510 din
86.	Parezanović: Računske mašine i progr. FORTRAN IV	450 din
87.	Muftić: Osnovni elementi kompjut. sistema	550 din
88.	Alagić: Principi programiranja	350 din

NOVO!

Ovih smo dana primili u prodaju i **najnoviju kasetu s 10 programa za zx Spectrum** koju je (s propratnim tekstom na srpskohrvatskom!) posle uspelog slovenačkog izdanja pripremila Software redekcija ljubljanskog Radia Študent.

Na odlično snimljenoj kaseti, koja košta **1300 din**, naći ćete same originalne programe mladih domaćih programera: igre, namenske i sistemske programe koji se mogu uporediti i s najboljim stranim programima (koji koštaju puno više)!

PORUČITE VEĆ DANAS — JER ĆE NAJTRAŽENIJE KNJIGE MOŽDA VEĆ SUTRA BITI RASPRODATE!

Sve knjige kao i računarsku kasetu za Spectrum možete poručiti pismeno ili telefonom (061/211-895) a posle-ćemo vam ih **pouzećem** (plaćete poštarinu prilikom prijema paketa). Za pravna lica važi obična narudžbenica. Ispunjenu narudžbenicu pošaljite na našu adresu:

KNJIGARNA MLADINSKE KNJIGE
61000 Ljubljana, Titova 3

SPECTRUMI ZA DINARE!

+6 originalnih kasata s igrama (16 K)
za škola i RO 54.073 din
za privatnika 69.700 din

Ostanimo u vezi

Časopis „Galaksija" u svakom broju objavljuje poseban blok tema posvećen kućnim računarima i njihovoj primeni u svakodnevnom životu. Sve ono što izdanjem „Računari u vašoj kući" nije bilo moguće obuhvatiti moći ćete da nađete u redovnim rubrikama u okviru tog tematskog bloka:

Pismo iz Londona

Ekskluzivni izveštaj sa vrućeg engleskog tržišta od specijalnog izveštača „Galaksije" iz Londona

Šta ima novo

Pregled novosti u svetu periferijskih jedinica, hardverskih dodataka, programa, časopisa i knjiga

Programeri u akciji

Novi konkurs za najbolji domaći program sa veoma primamljivim nagradama: ZX Spectrum 48K, računar „galaksija", ZX 81

Majstorije na računaru

Skrivene osobine računara „galaksija", ZX Spectrum i ZX 81 i kako ih iskoristiti

Računar „galaksija"

Programi i projekti hardverskih dodataka za samogradnju koji proširuju područja primene vašeg novog računara i povećavaju njegovu snagu

Servis računara

Odgovori na pitanja čitalaca o upotrebi, rukovanju, servisiranju i održavanju najpopularnijih modela računara

Bibiioteka bejzik programa

Najbolji domaći i strani komercijalni programi sa specijalno rađenim uputstvima i po simboličnim cenama

Mašinac za početnike

Popularna škola programira[nja] na mašinskom jeziku sa najva[...] nijim tajnama maternjeg jez[ika] računara

Softverska berza

Najveća berza za kupovinu, razmenu i prodaju originaln[ih] komercijalnih programa za [...] tipove računara — od TI 99 [...] Commodore 64



SPISAK NAREDBI SA PRIMERIMA

ARR$	ARR$(20)
BYTE	PRINT BYTE(11176) BYTE 11176,12
CALL	CALL 100 CALL 10*A+100
CHR$	PRINT CHR$(34)
DOT	DOT 20,10 DOT * IF DOT 20,10 PRINT "BELO"
EDIT	EDIT 20
ELSE	IF B<17 X=1: ELSE GOTO 100
EQ	IF EQ Y$,X$(5) GOTO 100
FOR	FOR I=A TO 100
GOTO	GOTO 100 GOTO 10*A+100
HOME	HOME HOME 64 HOME 512
IF	IF A>5 GOTO 100
INPUT	INPUT A INPUT X$
INT	A=INT(B/C)
KEY	IF KEY(1) GOTO 100 A=KEY(0)
LIST	LIST LIST 500
MEM	PRINT MEM
NEW	NEW NEW 256
NEXT	NEXT I
OLD	OLD OLD -50
PTR	PRINT PTR X$
PRINT	PRINT A+2*B PRINT A;"*";B;"=";C PRINT A,B,C PRINT "GALAKSIJA"
RETURN	RETURN
RND	A=RND PRINT INT(7*RND+1)
RUN	RUN RUN 100
SAVE	SAVE SAVE 15728,16251
STEP	FOR I=100 TO 0 STEP -5
STOP	STOP
TAKE	TAKE A,100,B,C,250,X$
UNDOT	UNDOT20,10 UNDOT *
USR	A=USR(14)
VAL	PRINT VAL(PTR X$)
WORD	WORD 10905,0 PRINT WORD(10905)
!	! PROGRAM ZA SUMIRANJE
#	#10,30,"JANUAR"
&	PRINT &2C8A

Uputstvo za upotrebu računar „galaksija"

Dejan Ristanović

Računar „galaksija" je, dakle, sastavljen i nalazi se pred vama. Teži i riskantniji deo posla je time završan — preostalo vam je da testirate kompjuter i — uživate u plodovima svoga rada! Vaš novi računar poznaje programski jezik bejzik i sa pravom isto to očekuje i od svog vlasnika. Da biste se, dakle, uspešno sporazumevali sa računarom, morate da uđete u tajne ovog jezika što, kao što ćete videti, nije mnogo težak posao. Ovo uputstvo za upotrabu će izložiti sve naredbe koje računar „glaksija" razume i dati primere njihove upotrebe. Ono, ipak, nije pisano tako da bude potpuno samostalno — u okviru ovog specijalnog izdanja nalazi se osnovna škola bejzika, koju bi svaki vlasnik računara „galaksija" koji nama iskustva sa programiranjem trebalo da pročita pre nego što nastavi sa ovim tekstom. Primetičete da su neke naredbe računara „galaksija" različite od onih koje su pomenute u školi bejzika. To ne treba da vas uplaši — bejzik nije standardizovan jezik i razlikuje se od kompjutera do kompjutera. No, kada se jednom potrudite i shvatite neki od njegovih dijalekata, prilagođavanje nekom drugom računaru je prava dečja igra.

Ovo uputstvo je koncipireno tako da vam omogući da pišete programe pre nego što ga potpuno pročitate i savladate. Zato su najvažniji oblici naredbi izloženi na njegovom početku, dok se deo za bolje poznavaoce računara nalazi na samom kraju. Onima koji imaju iskustva sa stonim računarima preporučujemo da jednostavno pogledaju spisak naredbi i memorijsku mapu i počnu sa radom — ne uputstvo će se vraćati kada osete potrebu.

1. Povezivanje sa televizorom

Računar „galaksija" može da bude povezan sa bilo kojim televizorom koji prima UHF područje. Za to se koristi koaksijalni priključak koji se nalazi sa njegove zadnje strane. Stoga uzmite odgovarajući kabl i povežite ga sa antenskim ulazom u televizor. Zatim uključite televizor, prebacite ga na UHF područje i laganim okretanjem birača kanala, pokušajte da na ekranu dobijete poruku READY (READY znači da je računar spreman da primi vaše naloge). Kada je ugledate, možete da uzviknete EUREKA — računar radi kako treba!

Čitaoci koji poseduju monitor ili prepravljeni televizor će poželeti da povežu računar sa njim i tako oslobode kućni TV aparat. Za to treba koristiti monitorski izlaz i kabl sa priključcima koji zavise od samog monitora.

2. Tastatura

Računar „galaksija" ima 54 testera koji su raspoređeni poput dirki na pisaćoj mašini. Kada god pritisnete neki od njih, računar na ekranu ispisuje odgovarajući simbol i to na mestu gde se prethodno nalazio kurzor (_). Čim se neki simbol pojavi na ekranu, kurzor se pomera za jedno mesto udesno da bi vam pokazao gde će se pojaviti sledeći znak. Ukoliko otkucate toliko slova da se ispuni čitav red, računar će automatski preći u sledeći bez ikakve crtice koja bi to označila — računar ne deli reči na slogove niti obraća pažnju na to da li je neka reč nelogično „presečena".

Pritisnite, na primer, taster A dvadesetak puta. Zatim će vam postati jasno da veliki broj slova A ne znači ništa, pa ćete poželeti da ih obrišete. Za to vam nije potrebno nikakvo korekturno mastilo — dovoljno je da pritisnete taster ← i kurzor će se pomeriti za


```
 90 C.3000
 100 C.2100
 170 B.P,43:IFKEY(27)D=-32
 180 IFKEY(28)D=32
 190 IFKEY(30)D=1
 200 IFKEY(29)D=-1
 210 IFKEY(48)@.1000
 300 B.P,32:A=P+D
 310 G.300+B.(A)
 332 P=A:@.170
 336 @.555
 342 H.:P.AT224,"UDARILI STE U Z
VEZDU I STRADALI!"!:G.1380
 396 D.555
 412 B.A,36:IFM=2@.170:E.N=N-1:I
F1-(N=0)G.170
 414 U.*:H.:P.AT202,"B  R  A  V
O"
 416 P.AT416,"VREME: ";Y$;
 418 @.1380
 555 D=-D:G.170
 1000 Y=P
 1010 IFKEY(31)G.170
 1020 IFKEY(27)X=-32:@.1070
 1030 IFKEY(28)X=32:@.1070
 1040 IFKEY(30)X=1:@.1070
 1050 IFKEY(29)X=-1:@.1070
 1060 G.1010
 1070 IFB.(Y)=46B.Y,32
 1080 Y=Y+X:G.B.(Y)+1100
 1132 B.Y,46:G.1070
 1136 IFM=2N=N-1:IFN=0@.414
 1138 B.Y,32:G.170
 1142 H.:P.AT224,"POGODILI STE Z
VEZDU I STRADALI!":G.1380
 1196 @.1138
 1212 H.:P.AT229,"POGODILI STE Z
IVU METU"!AT267,"I STRADALI!":@.
1380
 1355 H.:P.AT226,"POGODILI STE Z
ID I STRADALI!"
 1380 P.AT500,"JOS JEDNOM?";
 1382 IFKEY(4)@.1390
 1386 IF1-KEY(14)G.1382
 1388 IFKEY(14)G.1388:E.H.:S.
 1390 IFKEY(4)G.1390:E.@.100
 1400 #25,112,12,42,8,96,35,36
 1500 IFKEY(14)G.1500E.R.
 2100 H.:P."NIVO IGRE (1 ILI 2)"
;:I.M:IF1-(M=1)-(M=2)@.2100:E.IF
M=2C.6000
 2105 H.:T.0:F.I=10240TO10271:B.
I,255:B.I+480,255:N.I
 2110 F.I=10272TO10720S.32:B.I,2
55:B.I+31,255:N.I
 2120 F.J=1TO4:T.K,C:F.L=1TOK
 2130 I=INT(480*RND+10272):IF1-(
B.(I)=32)@.2130
 2140 B.I,C:N.L:N.J
 2150 P=INT(480*RND+10272):IF1-(
B.(P)=32)G.2150
 2160 F.I=53TO496.-1:B.P,I:F.J=1
TO120:N.J:N.I:D=0:N=25:IFM=2N=60
 2170 Y$="00:00:00":D.*:R.
 3000 H.
 3010 P.AT166,"*****************
**"
 3020 P.AT198,"* W O N D E R E R
*"
 3030 P.AT230,"*****************
**"
 3040 F.I=0TO63:D.I,0:D.I,47:N.I
 3050 F.I=1TO46:D.0,I:D.63,I:N.I
 3060 P.AT469,"UPUTSTVA? ";
 3070 IFKEY(14)@.1500
 3080 IF1-KEY(4)G.3070
 3090 H.:P."   U OVOJ AKCIONOJ I
GRI TREBA"
 3100 P."DA PRETVORITE SVE ZIVE
METE U"
 3110 P."MRTVE. ZIVA META POSTAJ
E MRTVA"
 3120 P."SAMO AKO UDARITE U NJU.
PORED"
 3130 P."ZIVIH I MRTVIH META NA
EKRANU"
 3140 P."CETE VIDETI ZVEZDE I  P
REPREKE"
 3150 P."(NA EKRANU JE SAKRIVENO
I NES-"
 3160 P."TO NEVIDLJIVIH PREPREKA
).MORA-"
 3170 P."TE DA PAZITE DA  NE  UD
ARITE U"
 3180 P."ZVEZDU DOK SE OD PREPRE
KA ODBI-"
 3190 P."JATE BEZ POSLEDICA."
 3200 P."   OPREMLJENI STE ORUZJ
EM KOJIM"
 3210 P."MOZETE DA UNISTAVATE PR
EPREKE"
 3220 P."(VIDLJIVE I NEVIDLJIVE)
ALI KO-"
 3230 P."JE NE SMETE DA UPERITE
U DRUGE"
 3240 P."OBJEKTE.";:C.4000
 3260 H.:P."OBJEKTI SU OBELEZENI
OVAKO:"
 3270 P.:P.:P."*   JE PREPREKA I
LI MRTVA META"
 3280 P.:P."@   JE ZIVA META"
 3290 P.:P."*   JE, NARAVNO, ZVE
ZDA"
 3310 P.:P."+   JE VAS POLOZAJ":
C.4000
 3350 H.:P."NA RASPOLAGANJU SU V
AM KOMANDE:"
 3360 P.:P.:P."STRELICE  ZA POME
RANJE"
 3370 P.:P."RET        ZA PUNJENJ
E ORUZJA"
 3380 P."            (ISPALJUJETE
GA PRI-"
 3390 P."            TISKOM NA STR
ELICU)"
 3400 P.:P."SPACE      AKO NAPUNI
TE ORUZJE"
 3410 P."            PA SE PREDOMI
SLITE."
 3420 P.:P.:P."    M N O G O   S
R E C E !"
 3430 C.4000:RET
 4000 P.AT497,"PRITISNI";
 4002 P.AT506,"RET";
 4004 IFKEY(48)G.4020
 4006 FORI=0TO60:N.I
 4008 P.AT506,"   ";
 4010 IFKEY(48)G.4020
 4012 FORI=0TO60:N.I:@.4002
 4020 IFKEY(48)G.4020
 4030 R.
 6000 H.:P."    NA OVOM NIVOU MOR
ATE DA UNI-"
 6010 P."STITE SVE STO SE NALAZI
NA EK-"
 6020 P."RANU OSIM ZVEZDA I (EVE
NTUALNO)"
 6030 P."NEVIDLJIVIH PREPREKA";:
C.4000:RET
```

31. *Sarađujmo i dalje*

Ovim smo stigli do kraja uputstva za upotrebu računara „galaksija". Ostalo je, čini nam se, još dosta stvari koje nismo pomenuli, „caka" koje ni mi ne poznajemo i, naravno, bezbroj ideja za programe. Ako ste sastavili računar „galaksija" i naučili da se služite njime, pošaljite nam bar kratko pismo u kom ćete nas obavestiti o tim činjenicama. „Galaksija" planira da otvori svoje stranice i posveti redovne rubrike temama vezanim za računar „galaksija", kao i da u postojeći katalog bejzik programa uvrsti što više dobrih programa za njega, koji će tako biti na raspolaganju svim korisnicima. Čitava ova akcija zavisi u mnogome i od svakog čitaoca ovih redova: vi ste ti koji treba da napišu programe i da ih pošalju na našu adresu. Samo udruženim naporima računar „galaksija" može da bude dopunjen onim što mu trenutno jedino nedostaje — programskom podrškom.



jedno mesto ulevo brišući automatski slovo na čije je mesto došao. Uzastopnim pritiscima na ovaj taster možete da obrišete sva otkucana slova — sve do početka reda.

Na tastaturi postoji, jasno, i tačka, ali ćete se za koji trenutak uveriti da vaš računar tu tačku ne smatra krajem rečenice. Otkucajmo, na primer, RACUNAJ i stavimo tačku. Kompjuter ne reaguje, što znači da nije ni shvatio da mu je izdata neka naredba. U programiranju je, naime, uobičajeno da se na kraju rečenice umesto tačke pritisne taster koji je označen sa **RET**. Pritisnimo ga i na ekranu će se pojaviti poruka WHAT? iza koje će slediti uobičajeno READY. Šta se dogodilo? Pritiskom na **RET** stavili smo računaru do znanja da je izdata naredba koju on treba da izvrši. On je analizirao tekst koji smo otkucali i primetio da se među naredbama koje poznaje ne nalazi RAČUNAJ. Zato je na ekranu ispisao WHAT?, a zatim i READY, stavljajući nam do znanja da je spreman za novu naredbu, koja, kako se nada, neće biti nerazumljiva.

Taster **RET** je, dakle, neobično značajan. Ne zaboravimo da ga pritisnemo kada god dovršimo neku naredbu ili rečenicu — ako to ne učinimo, računar će mirno čekati da nešto preduzmemo, ispoljavajući pri tom upornost koja daleko prevazilazi našu.

Ako nastavimo da pritiskamo RET, ekran će ubrzo biti popunjen. Tada će se prvi redovi postepeno gubiti, a ono što kucamo pojavljivati u poslednjem. Stalno opštenje sa poslednjim redom može nekome da se učini ponižavajućim — u tom slučaju treba da naredimo računaru da obriše sadržaj čitavog ekrana i počne da piše od njegovog vrha. Jedan od načina je da isključimo i ponovo uključimo računar, ali ima i jedan bolji.

Upoznajmo, najpre, taster **SHIFT.** Ovaj natpis ne vredi tražiti na tastaturi: odnosi se na dva neobeležena tastera. Istovremenim pritiskom na **SHIFT** i neki broj dobijamo na ekranu znak napisan iznad tog broja (pritisnimo, na primer, istovremeno **SHIFT** i 2 i na ekranu će da se pojavi znak navoda). Kod brojeva je, kao što vidimo, simbol napisan na samom tasteru ali postoje i neke dirke sa skrivenom šiftovanom funkcijom. Pritisnimo, na primer, istovremeno **SHIFT** i **DEL** (ukoliko smo, kao što je predviđeno pri montaži, predvideli dva **SHIFT** tastera, možemo da koristimo bilo koji od njih) i ekran će biti obrisan. Kada nam, dakle, računareve poruke WHAT?, koje su izazvane našim početničkim greškama dosade, pritisnućemo **SHIFT DEL** i obrisati ekran.

Taster **SHIFT** će nam otkriti još neke male tajne računara „galaksija". Pritisnimo, na primer, tastere **SHIFT** i C i na ekranu će se pojaviti slovo Č. Sasvim slično, **SHIFT** S daje slovo Š, **SHIFT** Z slovo Ž a **SHIFT** X (zapamtićete ga po tome što kapa iznad slova č ima oblik koji podseća na polovinu iksa) — Č. Ostali tasteri nemaju šiftovane funkcije — **SHIFT** A, na primer, daje slovo A.

Ako ste čitajući ovaj tekst eksperimentisali sa kucanjem verovatno ste namerno ili slučajno pritisnuli taster **REPT.** Tada se desilo nešto prilično čudno — računar je počeo da ispisuje gomile istih slova. Taster **REPT** dakle, ima funkciju koju programeri nazivaju „autorepeat" (automatsko ponavljanje) — pritisak na njega nalaže računaru da ponavlja funkciju poslednjeg pritisnutog tastera. Namerno nismo rekli „da ponavlja poslednje slovo": otkucajte red sa doste slova a zatim pritisnite←. Poslednje slovo je, kao što smo i očekivali, obrisano. No pritisak na **REPT** će proizvesti daleko efektnije dejstvo: računar će „gutati" slova sve dok ne obriše čitav red.

Osim tastere koje smo upoznali na tastaturi se nalaze još tri strelice kao i dirke sa tajanstvenim napisima **BRK** i **LIST/STOP** (u daljem tekstu **L/ST**). **BRK** je skraćenica od BREAK (u bukvalnom prevodu „slomi"), pritisak na ovaj taster nalaže računaru da prekine sa izvršavanjem nekog programa i ne ekranu ispiše READY. Taster **L/ST** nalaže računaru da na ekranu prikaže sadržaj programa koji se nalazi u njegovoj memoriji, a strelice nemaju neku posebnu funkciju — koriste se u igrama i računar može da kontroliše de li je neka od njih pritisnuta. Za sada ne umemo da pišemo programe pa su nam svi ovi tasteri prilično beskorisni, ali to neće trajati dugo.

Sa zadnje strane računara se nalazi dugme iznad koge bi trebelo da piše RESET. Ono predstavlja pojačanu verziju tastera **BRK**: ukoliko vam se ikada desi da se računar blokira i otkaže poslušnost (ovo se dešava, ako je računar ispravan, samo kod onih koji pišu programe na mašinskom jeziku ili pri čitanju programa sa kasete) pritisak na RESET će u 99% slučajeva izazvati ispisivanje uobičajenog READY, pri čemu memorija računara neće biti izbrisana. U preostalom procentu slučajeva moraćete da isključite i ponovo uključite računar, čime će programi biti izbrisani. Važno je zapamtiti da ni jednom naredbom ili grupom naredbi računar ne može da bude oštećen — isključivanje i ponovno uključivanje će ga uvek povratiti.

Pre nego što pođemo dalje moramo da izreknemo i jedno upozorenje vezano za dugme RESET: ono nije namenjeno stalnoj upotrebi. Pokušajte na primer da brzo pritiskate RESET više puta, pa lako može da se desi da memorija računara bude kompletno izbrisana. RESET, dakle, koristite samo kao poslednju soluciju, kada nikako drukčije ne možete da „probudite" vaš kompjuter.

3. Demo kaseta

Uz računar „galaksija" ste svakako naručili i kasetu sa programima. Ona je namenjena testiranju računara, navikavanju na rad sa kasetofonom i prevashodno zabavi. Da biste je koristili považite džek na kome piše EAR (ili neki drugi napis, zavisno od kasetofona) sa računarom prema ranije prikazanoj šemi, stavite demo-kasetu, otkucajte OLD, pritisnite **RET** i startujte kasetofon. Ovo OLD, doduše, znači star ali neće naterati računar da postane stara i istrošena mašina — ono mu jednostavno nalaže da sa kasete učite program koji je ranije snimljen i pripremi ga za izvršavanje. Ekran će se najpre potpuno zatamniti da bi se desetak sekundi docnije pojavilo READY. Ako se na ekranu i posle tridesetak sekundi ne pojavljuje ništa, računar nije primio program. Bez panike — to se dešave zbog toga što jačina zvuka ne odgovara potrebama kompjutera. Pritisnite zato RESET, premotajte kasetu na početak, ponovo otkucajte **OLD RET** i startujte kasetofon uz malo manipulisanje sa potenciometrom za regulisanje jačina zvuka. Posle kraćeg ili dužeg truda program će biti učitan u memoriju računara i moći ćete da ga startujete. Startovanje se obavlje kucanjem naredbe RUN i pritiskanjem tastera **RET**. Bez brige: računar neće nikuda pobeći (RUN=trči, beži) nego će početi sa izvršavanjem upravo učitanog programa. **Ne** možemo da vam kažemo šte će se tada desiti. U okviru prvog programa računar će vas upoznati sa sadržajem demo-kasete i poučiti šta da radite.

Za učitavanje preostalih programa procedura je ista: otkucate OLD pritisnete **RET**, startujete kasetofon i kada se na ekranu pojavi READY otkucate RUN i pritisnete **RET**. Računar će zatim ispisati ime programa i upitati vas da li su potrebna uputstva za njegovu upotrebu. Odgovorite sa D (od DA) i pažljivo ih pročitajte.

30. Potprogrami iz ROM-a

Čak i dok niste započeli da pišete programe, u vašem računaru se nalazio jedan program i to toliko veliki i moćan da ćete sličan moći da napišete tek kad postanete daleko iskusniji programer. To je sistemski program od četiti kilobajta koji omogućava računaru „galaksija" da radi ono što mu vi nalažete: taj program omogućeve prevođenje bejzik naredbi koje unosite na oblik pristupačan mašini, izdavanje odgovora na ekranu, rad sa kasetofonom...

U jednom velikom programu, kao u metrou, ima mnogo stranica na kojima možete da se „ukrcate", otputujete negde i „izađete". Poznevanje adresa tih „stanica" je dovoljno da iskoristite naredbu USR i dođete do njih. Vreme koje nam je bilo na raspolaganju za pripremanje ovog uputstva kao i njegov obim nam, na žalost, nisu dopustili da se pozabavimo nabrajanjem interesantnih mašinskih potprograma. Odabrali smo samo jedan primer, koji će čitaoca podstaći na dalja istraživanja i istovremeno povećati mogućnosti računara „galaksija".

U toku rada (osim pri snimanju programa na kasetu) računar na ekranu neprekidno prikazuje sliku, na šta se prirodno gubi određeno vreme. Ponekad nam je potrebno da izvršimo neki komplikovani proračun koji odnosi dosta vremena, a slika nam nije bitna, zer ne bi bilo dobro „isključiti" sliku i smanjiti vreme izvršavanja programa tri-četiri puta? Za ovakav ubrzani režim rada poslužiće nam naredba A=USR(14) (umesto A možemo da stavimo bilo koju promenljivu, napišemo PRINT USR(14) i slično). Povratak u normalan režim rada se postiže naredbom A=USR(22) ili automatski nastupa po završetku rada programa (taj završetak može da bude posledica naredbe STOP, pritiska na BRK ili RESET i slično). Evo i primera:

```
10 A=USR(14)
20 FOR I=1 TO 10000  NEXT I
```

Startujmo program sa RUN a zatim sa RUN 20 (time ubrzani režim nije iniciran) i primetimo razlike u vremenu izvršavanja. Za to ne možemo da koristimo računarev interni časovnik jer je on „isključen" u toku ubrzanog režima rada.

```
5 PRINT
10 FOR A=32 TO 95
20 PRINT CHR$(A);
30 NEXT A
40 FOR B=128 TO 191
50 PRINT CHR$(B);
60 NEXT B
80 GOTO 80
READY
>RUN
```

Kompjuterska abeceda slova koja je računar „galaksija" „naučio"

Imate li dobre reflekse. Jedna od mnogobrojnih akcionih igara prilagođena računaru „galaksija"

```
10 INPUT X: CALL 30000:PRINT Y
20 GOTO 10

30000 ! KVADRATNI KOREN
30010 IF X=0 Y=0: RETURN
30020 IF X<0 PRINT "KOREN NIJE
DEFINISAN": STOP
30030 Y=X*.5: Z=0
30040 W=(X/Y-Y)*.5
30050 IF (W<0) + (W=Z) RETURN
30060 Y=Y+W: Z=W: GOTO 30040
READY
>RUN
?2
 1.41421
?55
```

```
10 INPUT X: CALL 31000
20 PRINT"LN(X)=";L,"LOG(X)=";Y
30 GOTO 10
31000 ! LOGARITAM
31010 E=0: IF X<0 PRINT "FUNKCI
JA NIJE DEFINISANA": STOP
31020 IF X<1 ELSE X=X/2: E=E+1:
GOTO 31020
31040 IF X<0.5 X=X+X: E=E-1: GO
TO 31040: ELSE X=(X-0.707107)/(X
+0.707107): L=X*X
31050 L=(((0.598979*L+0.961471)
*L+2.88539)*X+E-0.5)*0.693147
31060 IF (L<1E-6)*(L>-1E-6)L=0
31070 Y=L*0.4342945: RETURN
```

Za matematički nastrojene: Potprogram ... za kvadratni koren ... i logaritam

Mašinski program se unosi tako što se najpre otkuca NEW 15 (program ima 15 bajtova), a zatim BYTE&2C3A, &01 **RET** BYTE&2C3B, &00 **RET** BYTE&2C3C, &02 **RET** BYTE&2C3D, &21 **RET** itd. Poslednje se kuca BYTE&2C47 &F8 **RET** BYTE &2C48 &C9 **RET** čime je unošenje mašinskog programa završeno. Da bismo ga isprobali, otkucajmo

```
10 A=USR (&2C3A)
20 GOTO 20
```

USR je nova naredba. Ona nalaže računaru da izvrši mašinski program čiji se početak nalazi na adresi koja je navedena u zagradi. Po izvršavanju programa(mašinsko RET, C9) promenljiva A će dobiti vrednost iz HL registra, što u datom primeru nije mnogo bitno. Naredba 20, jednostavno, sprečava da se izvršavanje programa prekine i da izgled ekrana tako bude pokvaren.

Otkucajmo RUN, pritisnimo **RET** i, ako nije bilo grešaka u kucanju, ekran će biti popunjen tačkama. Uporedimo ovo sa „brzinom" (posle ovakve demonstracije navodnici su neophodni) BEJZIK programa koji smo nedavno sastavili pa će nam postati jasno zašto se bolji programeri okreću mašinskom jeziku. Za one koji ne mogu da se sete kako da izmere vreme izvršavanja mašinskog programa da kažemo da je ono oko 0.02 sekunde što, u poređenju sa bejzikom, predstavlja ubrzanje od oko 580 puta.

Želeli bismo da maksimalno stimulišemo pisanje programa na mašinskom jeziku. Računar „galaksija" je dobar poligon za njih: dugme RESET će popraviti posledice najvećeg broja grešaka, tako da nećete morati stalno da unosite program kada računar „krahira". I pored toga, biće neophodno da mašinske programe snimate na kasetu. U tome će vam pomoći sledeće poglavlje.

29. Nove mogućnosti naredbi SAVE i OLD

Ako mašinski program unesemo na način izložen u prethodnom poglavlju, obično SAVE će, zajedno sa bejzikom, snimiti i sve što se nalazi u memoriji, počevši od adrese &2C3A. Ponekad je ipak potrebno da na kasetu snimimo proizvoljan segment memorije. Za to će nam poslužiti naredba SAVE mmmm, nnnn.

Da bismo, na primer, snimili mašinski program koji se nalazi u ćelijama memorije čije su adrese &3C30 do &3FFF, otkucaćemo SAVE &3C30, &3FFF, startovati snimanje na kasetofonu i pritisnuti **RET.** Posle toga, ukoliko nam je program važan, možemo da izvršimo verifikaciju na uobičajeni način.

Za učitavanje koristimo obično OLD: računar će učitane bajtove smestiti na iste adrese na kojima su se nalazili pre snimanja. Ponekad ćemo, međutim, poželeti da učitavanje programa započnemo određen broj bajtova pre ili posle mesta sa koga su snimljeni (ovo retko ima smisla jer će sve JP i CALL mašinske naredbe neispravno funkcionisati posle pomeranja). Ako, na primer, želimo da mašinski program bude pomeren naviše za 1024 bajta, otkucaćemo OLD 1024 (ili, što je isto, OLD &400), pritisnuti **RET** i startovati kasetofon. Da je pomeranje trebalo da bude negativno (prema početku memorije) otkucali bismo OLD — 1024. Obzirom da je ovakvo pomeranje relativno, svaki put kada snimamo neki mašinski program pomoću SAVE mmmm, nnnn, treba da zapišemo adresu njegovog početka i kraja.

Znanje je sila, znanje je moć   učite deco i dan i noć

4. Komandni i programski mod

Možda će se naći neko ko dalji tekst neće ni čitati — nekoliko igara sa demo-kasete će mu oduzeti toliko vremena da će smatrati da je računar ispunio svoju funkciju i opravdao uloženi novac. Ipak, učenje programiranja je lak i prijatan posao u koji se vredi upustiti.

Počećemo na najjednostavniji mogući način: otkucaćemo slovo po slovo tekst **PRINT** "RAČUNAR GALAKSIJA" i pritisnuti **RET** (navodnici se dobijaju istovremenim pritiskom na tastere **SHIFT** i 2, a slovo Č — pritiskom na **SHIFT** X). Tako smo naložili računaru da na ekran prenese tekst naveden između znakova navoda — u ovom slučaju **RAČUNAR** GALAKSIJA. Možemo da ponavljamo ovu naredbu veći broj puta umećući između znakova navoda bilo koji tekst proizvoljne dužine i računar će biti poslušni izvršilac **naših želja**. Svaka naredba se analizira i izvršava onoga momenta kada smo pritisnuli **RET**, pa se ovaj način rada naziva **komandnim** (na ovaj način računaru izdajemo komande kojima kontrolišemo njegov rad).

Osim komandnog treba da upoznamo **programski** režim rada. I kod njega računaru izdajemo komande, ali se one ne izvršavaju odmah — računar ih pamti i izvršava tek kada mu to bude naloženo. Da bi kompjuter razlikovao komande od naredbi koje treba da „pamti", uvedeni su brojevi programskih linija. Broj programske linije (ili, kao što se ponekad naziva, linijski broj) ukazuje računaru na redosled kojim će, kada mu to bude naloženo, izvršavati „zapamćene" instrukcije. Ovaj broj treba da se nalazi na samom početku linije.

Otkucajmo, na primer, 10 PRINT "RAČUNAR GALAKSIJA" i pritisnimo **RET**. Računar nije izvršio naredbu: ona je zapamćena i dodeljen joj je linijski broj 10. Otkucajmo zatim 20 PRINT "JE MOJ PRVI KOMPJUTER" i ponovo pritisnimo **RET**. Obzirom da je linijski broj druge naredbe (20) veći od linijskog broja prve (10), računar će najpre ispisati RAČUNAR GALAKSIJA a zatim JE MOJ PRVI KOMPJUTER. Da proverimo ovo rezonovanje, otkucaćemo RUN (ovo je naredba koju smo dobro upoznali igrajući se sa demo kasetom — ona nalaže računaru da započne sa izvršavanjem programa) i pritisnuti neizbežno **RET**. Računar će, kao što se i očekivalo, izvršiti program i na ekranu ispisati

```
RAČUNAR GALAKSIJA
JE MOJ PRVI KOMPJUTER
READY
```

Ono READY nije posledica neke od naredbi našeg programa već uobičajeni signal da je računar spreman da primi sledeću naredbu. To ponovo može da bude RUN — program će se izvršavati onoliko puta koliko poželimo.

Pokušajmo da proširimo program koji smo sastavili i da se uverimo da računar izvršava naredbe po redosledu njihovih brojeva a ne po redu unošenja: otkucajmo 15 PRINT "KONSTRUISAN SEPTEMBRA 1983" i pritisnimo **RET**. Po startovanju programa (RUN i **RET**) na ekranu će biti ispisan tekst

```
RAČUNAR GALAKSIJA
KONSTRUISAN SEPTEMBRA 1983
JE MOJ PRVI KOMPJUTER
READY
```

što znači da je linija 15 umetnuta između linija 10 i 20 — baš kao što smo i očekivali.

Ako na ovaj način počnemo da umećemo naredbe, ubrzo ćemo zaboraviti kako program izgleda. To je lako proveriti: jednostavno pritisnemo taster na kome je ispisano **LIST** (LIST bi slobodno moglo da se prevede sa „prikaži program"). Dok taster **LIST** držimo pritisnut, računar će ispisivati program koji se nalazi u njegovoj memoriji liniju po liniju. Čim pustimo ovaj taster, ekran će se „zamrznuti", pa ćemo moći da analiziramo uneseni program. Ako želimo da se listanje nastavi, ponovo ćemo pritisnuti taster **LIST**, a ukoliko ne želimo pritisak na BRK će na ekranu prikazati uobičajeno READY.

Umesto da pritiskamo taster **LIST**, možemo da otkucamo reč LIST i pritisnemo **RET**. Sve dok je **RET** pritisnut, prikazivanje programa traje, a otpuštanje tastera privremeno prekida ovaj proces. Na prvi pogled nema smisla koristiti naredbu LIST kada je obezbeđen taster sa istim dejstvom. Ovakav zaključak je samo donekle tačan: ponekad ćemo poželeti da prikazujemo program počevši od neke linije. U tom slučaju treba da

otkucamo LIST, a zatim navedemo broj te linije i pritisnemo **RET** (na primer, LIST 200). Na taj način ne moramo da posmatramo početak programa ako nam je potreban njegov kraj. Za početak će, doduše, naši programi biti toliko kratki da će bez problema stajati na ekran, pa naredbu LIST ostavljamo za blisku budućnost.

Ni izvršavanje programa ne mora da počne od početka: možemo na primer da otkucamo RUN 15 i pritisnemo **RET.** Računar će početi da izvršava program od linije 15 i ispisati na ekranu

KONSTRUISAN SEPTEMBRA 1983
JE MOJ PRVI KOMPJUTER
READY

Ova rečenica je naravno besmislena. No za računar nije postojao nikakav razlog da odbije da je prenese na ekran, s obzirom da su sve instrukcije u programu bile valjane. U budućnosti ćemo videti da postoje programi koji ne mogu da se izvršavaju od proizvoljne linije bez katastrofalnih posledica (računajući i gubitak čitavog programa) ali nam je i ovo bila dobra škola — započinjanje programa od neke neplanirane tačke ne mora da izazove grešku ali su rezultati nepredvidljivi.

Unošenje programa i njegovo neprekidno izvršavanje je bez sumnje lepa zabava ali će nam i ona dosaditi. Kada poželimo da izbrišemo otkucani program iz memorije, možemo da isključimo računar iz napajanja i trenutak docnije ga ponovo uključimo ili, što je sasvim isto, otkucamo PRINT USR (0) i pritisnemo **RET.** Na raspolaganju je i naredba NEW (posle nje se, naravno, pritiska **RET**) koja briše samo program, ali ne i sadržaj ekrana i promenljivih.

5. „Galaksija" kao kalkulator — PRINT, INT

Naredbe PRINT koje smo do sada koristili su nalagale računaru da tekst otkucan između navodnika jednostavno prenese na ekran. No, ova naredba može da obavi i mnogo aktivniju ulogu. Otkucajmo, na primer, PRINT 17+13 i pritisnimo **RET.** Na ekranu će se pojaviti broj 30, što znači da je postavljeni zadatak izvršen! Umesto 17+13 možemo da otkucamo bilo koji izraz na način koji je uobičajen u matematici, uz upotrebu zagrada, pri tome treba da pazimo da se množenje obeležava zvezdicom (*) a deljenje kosom crtom (/) i da se množenje ne podrazumeva. PRINT (12+2)(17+3) bi izazvalo grešku dok bi se PRINT (17+2)*(17+3) korektno izvršilo. Evo, na primer, kako bi izgledala naredba koja izračunava brojni izraz

$$3 \cdot \frac{4.5+17.2 \cdot 19.45}{2.1-12.4/16.22}$$

PRINT 3*(4.5+17.2*19.45)/(2.1−12.4/16.22)) **RET**

Izvršavanje ove naredbe će na ekranu dati rezultat 761.595, koji je tačan na šest cifara.

Videli smo da brojevi sa kojima računar „galaksija" operiše ne moraju da budu celi — dovoljno je da koristimo decimalnu tačku umesto kod nas uobičajenog zareza. Ipak, brojevi sa kojima možemo da radimo su ograničeni na raspon od −999999 do 999999. Brojevi veći ili manji od ovih limita se prikazuju na drugi način: pomoću stepena broja 10. Otkucajmo, na primer, PRINT 1000000 i računar će ispisati 1E+06 (podrazumevamo da ste zapamtili da se posle svake naredbe pritiska **RET,** pa ubuduće nećemo pisati odgovarajući komentar). Slovo E potiče od reči eksponent, a zapis se prevodi kao $1 \cdot 10^6$. Podsećanje na osnovačku matematiku će nas uveriti da je $1 \cdot 10^6$ isto što i 1000000 — jedino je zapis drukčiji. Čak ni na ovaj način ne možemo da operišemo sa neograničeno velikim brojevima — ukoliko neki od argumenata ili rezultat neke operacije bude manji od 10^{-38} ili veći od 10^{37}, računar će na ekranu ispisati HOW? — tražili smo od njega nešto što prevazilazi njegove računske sposobnosti.

Krajnje je vreme da sastavimo i prvi program. Obzirom da tek počinjemo, to će biti jednostavan programčić koji računa vreme potrebno za putovanje od jednog do drugog

Međutim, većina drugih sistemskih promenljivih zauzima dva bajta obzirom da se u njima nalaze adrese nekih drugih delova memorije (sistemska promenljiva koja zauzima ćelije &2C36 i &2C37, na primer, daje adresu početka bejzik programa). To znači da nam naredba BYTE pomaže samo posredno: možemo da ispitamo sadržaje obe ćelije koje čine neku sistemsku promenljivu i da ih kombinujemo u jedan broj (u datom primeru PRINT 256*BYTE (&2C37)+BYTE (&2C36) daje 11322 (isto što i &2C3A) — adresu početka bejzik programa koja se slaže sa mapom). Ovakav način, posebno kada treba promeniti sadržaj neke sistemske promenljive, nije baš prijatan pa je „galaksija" opremljena naredbom WORD koja je namenjena baš radu sa sistemskim promenljivima.

Naredba WORD je neobično slična naredbi BYTE osim u jednoj sitnici: ona operiše sa dva bajta memorije. Otkucajmo, na primer, PRINT WORD (&2C36) i na ekranu će se pojaviti broj 11322 — nikakva konverzija nije bila potrebna.

Otkucajmo dalje: WORD &2C36, &2D3A: WORD &2C38, &2D3A i pritisnimo **RET.** Time smo promenili sadržaj sistemskih promenljivih u kojima računar čuva informacije o početku i kraju bejzik programa, pa će svi bejzik programi koje od tog momenta kucamo počinjati 256 bajta više (počevši od &2D3A). Tako dobijenih 256 bajta može, na primer, da se koristi za smeštanje mašinskih programa.

28. Mašinski programi — naredba USR

Raspoloživi prostor nam, na žalost, ne dopušta da se ozbiljno bavimo mašinskim programiranjem, ali ćemo izložiti neke podatke vezane za smeštanje, snimanje i startovanje mašinskih programa na računaru „galaksija" i dati jedan primer. Za praćenje daljeg teksta je vrlo korisno posedovati spisak instrukcija procesora Z80A.

Postoje dva mesta koja su „prirodno određena" za mašinske programe: jedno je već spomenuto (ispred bejzika), a drugo na vrhu memorije, tamo gde se normalno nalazi niz X$(i). Prva ideja je, bar po našem mišljenju, pogodnija jer se mašinski programi snimaju zajedno sa bejzikom (o tome docnije), pa ćemo se **zadržati** na njoj.

Da bi za mašinski program rezervisali **nnnn** bajtova, treba da pomerimo početak bejzika. To možemo da uradimo pomoću naredbe WORD (kao u prethodnom poglavlju) ali možemo da upotrebimo i novu opciju NEW nnnn. Primenom ove naredbe briše se postojeći bejzik program iz memorije i početak svih budućih bejzik programa pomera za nnnn bajtova. Mašinski program se, u nedostatku programa koji bi to olakšao, unosi pomoću naredbe BYTE, bajt po bajt počevši od &2C3A. Da vidimo kako to izgleda na primeru.

Sastavićemo program koji popunjava ekran punim karakterima i videti koliko je mašinski jezik brz. Evo programa uz korišćenje standardne mnemonike procesora:

```
2C3A 01 00 02        LD BC, 200H      ; treba „osvetliti" &200 bajtova
2C3D 21 00 28        LD HL, 2800H     ; adresa početka ekrana
2C40 36 FF   CIKL    LD (HL), FFH     ; FF je „pun karakter"
2C42 23              INC HL           ; HL se uvećava
2C43 0B              DEC BC           ; brojač je umanjen
2C44 CB 78           BIT 7, B         ; da li je BC negativan?
2C46 28 F8           JR Z, CIKL       ; ako nije, ponavlja se ciklus
2C48 C9              RET              ; povratak u BASIC
```

Pomoć konstruktoru: računar projektuje astabilni multivibrator

Sledeći potez je vaš: Popunite ovu blok-šemu

dosta neprijatnih trenutaka. Da bi se problemi rešili, treba upoznati naredbu PTR. Izvršimo, na primer PRINT PTR X$ i na ekranu će se pojaviti broj 10864. To znači da je promenljive X$ u memoriji smeštena počevši od ćelije 10864 (ili, što je isto, &2A70). Znajući da alfanumerička promenljiva prima 16 slova, zaključujemo da će ćelije &2A70 do &2A7F koriste za promenljivu X$. Unesimo sledeći jednostavni program

```
10 INPUT X$
20 Y$=""
30 FOR I=0 TO 2
40 BYTE PTR Y$+I, BYTE (PTR X$+I)
50 NEXT I: BYTE (PTRY$+3), 0
```

Startujmo ovaj program i unesimo bilo koju reč koja ima manje od 16 slova. Trenutak docnije na ekranu će se pojaviti prva tri slova reči. Čitaoci ovih redova koji su razumeli rad sa alfanumericima će primetiti da bi naredbe 40 i 50 mogle da se zameni sa

```
40 Y$=Y$+CHR$ (BYTE (PTR X$+I): NEXT I
```

Naredbu PTR možemo da koristimo i za pronalaženje elemenata numeričkog ili alfanumeričkog niza ili fiksnih promenljivih. Kao vežbu pokušajte da sastavite program koji će sortirati reči po abecednom redu. Problem postaje bitno složeniji kada posmatramo reči koje sadrže i slova Š, Ž, Č i Ć.

27.
Sistemske promenljive — naredba WORD

Tabela na slici prikazuje mapu sistamskih promenljivih i drugih značajnih memorijskih adresa. Sistamske promenljive označavaju memorijske ćelije u koje računar smešta podatke od značaja za njegovo normalno funkcionisanje. Jednu od sistemskih promenljivih smo već upoznali — ćelija čija je adrssa &2BA8 odgovorna za poziciju slike na ekranu.

SISTEMSKE ADRESE

ADRESA	BAJTA	INICIJALNO	SADRŽAJ
2800	512	20	VIDEO MEMORIJA
2A00	104	00	NUMERICKE VARIJABLE A-Z
2A68	2	2800	POZICIJA KURZORA U MEMORIJI
2A6A	2	3800	KRAJ MEMORIJE
2A6C	2	00	BLOKIRANJE DELA VIDEO-MEMORIJE
2A6E	2	00	>TO< REGISTAR ZA FOR-NEXT
2A70	16	00	X$
2A80	16	00	Y$
2A91	2	00	STEP ZA AKTIVNU PETLJU
2A93	2	00	POZICIJA TEKUCE LINIJE (ZA VREME FOR-NEXT)
2A95	2	00	BASIC POINTER (ZA VREME CALL I FOR-NEXT)
2A97	2	00	16*ARN0+16
2A9B	2	00	ADRESA NEXT VARIJABLE
2A9D	2	2C3C	TAKE POINTER
2A9F	2	00	POZICIJA TEKUCE LINIJE
2AA1	2	00	REGISTAR ZA AKTIVNI FOR-NEXT
2AA3	2	00	SP PRIVREMENO ZA AKTIVNI CALL
2AA5	2	00	DIFERENCIJATOR ZA TASTATURU
2AA7	3	00	RND
2AAC	124	00	ARITMETICKI AKUMULATORI (IX)
2BA8	1	11	HORIZONTALNA POZICIJA TEKSTA
2BA9	3	C9	LINK ZA NAREDBE
2BAC	3	C9	LINK ZA VIDEO
2BAF	1	00	AKO JE BIT 7=1, SAT RADI
2BB0	1	00	BROJAC ZA POMERANJE SLIKE
2BB1	1	00	FLAG ZA POMERANJE SLIKE
2BB1	1	00	REGISTAR ZA REPT
2BB5	1	00	FLAG ZA STAMPAC
2BB6	125	00	BAFER
2C36	2	2C3A	POINTER POCETKA BASICA
2C38	2	2C3A	POINTER KRAJA BASICA
2C3A	??	00	BASIC PROGRAM

mesta. Da pojednostavimo problem, pretpostavićemo da je brzina kretanja konstantna na čitavom putu i da je unapred zadata (ovakvim zahtevima bi se verovatno pokoravao voz koji ne nailazi na semafore, krivine i slične prepreke). Ako put koji trebe da se pređe označimo slovom s, brzinu slovom v a potrebno vreme slovom t, važiće relacija t = s/v (ovu relaciju osnovci uče u VI razredu a i mladima bi trabala da bude očigledna: što je veći put, duže se putuje, što je veće brzina, kraće se putuje). No pre nego što počnemo da pišemo program, pokušaćemo da iskoristimo naredbu PRINT i proverimo kako sve u praksi izgleda.

Od Beograda do Dubrovnika, na primer, ima 370 km u pravoj liniji, a avion „Caravelle" se kreće brzinom od nekih 800 km/h. Da nađemo vreme (u časovima) otkucaćemo

PRINT 370/800 i na ekranu će se pojaviti broj 0.4625. Olovka i papir (za sada još ne umemo da iskoristimo računar da obavi sve potrebne operacije) će pokazati da ovo predstavlja 27 minuta i 45 sekundi.

Koliko bi put trajao da se avion kretao brzinom od 900 km/h ili, šta da je putovao od Beograda do Zagreba? Mogli bismo, jasno, stalno da kucamo naredbe PRINT ali ćemo uvideti da od toga nema velikih vremenskih ušteda. Zato ćemo u program uneti instrukciju

30 PRINT S/V

30 je, kao što se sećamo, broj programske linije ali šta su S i V? Programeri ih nazivaju **promenljive** i korista za smeštanje nekih podataka u memoriju računara. Svaka promenljiva je ime nekog podatka koji docnije može da bude pozivan da bi se sa njim računalo. Njeno ime može da bude bilo koje slovo od A do Z (ne računajući slova Č, Ć, Ž i Š) i pogodno je koristiti slova koja asociraju na podatak koji promenljiva čuva (mi smo upotrebili slova S i V). Ipak, program koji smo sastavili nije potpun. Ako otkucamo RUN na ekranu će da se pojavi broj 1 koji svakako ne predstavlja rešenje našeg problema. Razlog je jednostavan: računar „galaksija" još nije naučio da čita naše misli i nema nikakav način da zna šta bi trebalo da sadrže promenljive S i V. U početku rada one imaju jednaku vrednost (**pažnja**: ta vrednost nije nula!) pa je rezultat deljenja jedan. Zato ćemo da dodamo naredbe 10 S=370 i 20 V=800 (nismo zaboravili da se posle svake linije pritiska **RET?)** i startovati program sa RUN. Ovoga puta sve je korektno radilo i računar je ispisao vreme putovanja od Beograda do Dubrovnika i to izraženo u časovima.

Ako želimo da ubrzamo zamišljeni avion otkucaćemo 20 V=900. Kako nova linija ima isti linijski broj kao i stara, jedna od njih mora da napravi mesto drugoj. Računar uvek smatra da je linija koju smo poslednju otkucali merodavna i uklanja iz memorije liniju 20 V=800 da bi je zamenio sa 20 V=900. Program se ponovo startuje sa RUN i dobija se novi rezultat.

Prikazivanje rezultata koje smo primenili nije baš srećno izabrano: računar nam prikazuje vreme u časovima i to na način koji nam nije prirodan: navikli smo da se vreme izražava u časovima, minutima i sekundama. Da prikažemo rezultat na ovakav način potrebna nam je jedna nova funkcija — INT. INT je skraćenica od INTEGER i označava, u bukvalnom prevodu, ceo broj. U „galaksijinom" bejziku INT je funkcija koja izračunava „ceo deo" izraza koji je naveden iza nje. Ako, na primer, otkucamo PRINT INT (3.1415) i pritisnemo **RET**, na ekranu će se pojaviti broj 3 — računar je jednostavno odbacio decimale broja u zagradi. U zagradi nije morala da se nalazi konstanta: mogli smo da napišemo PRINT INT (A/1000) ili čak PRINT INT (2,5+(A*17.2+C)).

Da bismo broj pretvorili u časove, minuta i sekunde, moramo da pomnožimo vreme u časovima sa 60 i ceo deo rezultata proglasimo za potreban broj minuta, broj časova dobijamo deljenjem vremena sa šezdeset i uzimanjem celog dela rezultata (možete da uzmete papir i olovku i proverite ovo rezonovanje na primerima: nema nikakvog smisla da nastavljate sa čitanjem pre nego što vam do sada korišćene naredbe ne postanu kristalno jasne). Evo i programa:

```
10 S=370
20 V=800
30 T=S/V
40 PRINT INT(T)
```

```
50 PRINT ,,ČASOVA,'
60 M=INT(M*60)
70 PRINT M
80 PRINT  MINUTA
```

Program, kao što se vidi, prikazuje samo časove i minute ali ćete, ako ste ga razumeli, bez problema povećati tačnost navođenjem sekundi!

6. Naredba INPUT

Da bismo u prethodnom programu izbegli neprekidno kucanje brojnog izraza koji računa s/v i koma prethodi naredba PRINT, koristili smo promenljive. Na žalost, ni na ovaj način kucanje nije bitno skraćeno, bolje bi bilo da nas, pri svakom izvršavanju programa, računar pita za pojedine vrednosti, a da mi samo kucamo potrebne brojeve. Ovaj problem rešava naredba INPUT, čije ćemo dejstvo upoznati pošto modifikujemo prethodni program

```
10 PRINT ,KOLIKO SU MESTA UDALJENA?'
15 INPUT S
20 PRINT ,KOLIKA JE BRZINA?'
25 INPUT V
```

Kada računar naiđe na naredbu INPUT, privremeno će da prekine sa izvršavanjem programa i na ekranu ispiše znak pitanja. Od korisnika se očekuje da otkuca neki broj i kada to učini, pritisne **RET**. Računar će otkucani broj dodeliti promenljivoj čije je ime navedeno iza INPUT i nastaviti sa radom.

Program se, kao i obično, startuje sa RUN. Računar ispisuje KOLIKO SU MESTA UDALJENA?

?_

Otkucajmo broj 370 i pritisnemo **RET.** Računar ispisuje KOLIKA JE BRZINA?

?_

Čim otkucamo 800 i pritisnemo **RET**, na ekranu će pisati:

```
0
ČASOVA
27
MINUTA
READY
```

READY je znak da je računar spreman za dalji rad: možemo da otkucamo RUN i damo podatke za neki drugi grad ili brzinu. No, ubrzo će nam dosaditi stalno kucanje naredbe RUN. Zar ne bi bilo bolje naložiti računaru da se, po izvršavanju programa i izdavanju rezultata, vrati na početak programa i od nas zatraži nove podatke? Za to će nam dobro poslužiti naredbe bezuslovnog prelaska (skoka).

7. Naredbe bezuslovnog i uslovnog prelaska — GOTO, IF . . ELSE, STOP

Dodajmo našem programu naredbu 90 GOTO 10 (pažnja: u principu nije bitno da li u programu ostavljamo prazne prostore između GOTO (ili bilo koje druge naredbe) i broja 10 ali **ne smemo** da ostavljamo prostor unutar same naredbe. GO TO 10 ća izazvati

0000 ROM A 0FFF		NA PLOČI
1000 ROM B 1FFF	PRAZNO MESTO	
2000 tastatura 2037		
2038 latch 203F		
2040 PONAVLJA SE LATCH I TASTATURA JOŠ 15 PUTA 27FF	NEUPO-TREBLJIVO	
2800 RAM C 2FFF	2800-2BFF video ram	
3000 RAM D 37FF		
3800 RAM E 3FFF		
4000 NIJE KORIŠĆENO NA PLOČI RAČUNARA FFFF	ZA PROŠI-RENJA	

25. Organizacija memorije — naredba BYTE

Naš računar je opremljen ROM-om od četiri kilobajta i istim tolikim RAM-om (dodavanjem čipova ROM može da se proširi do 8 Kb a RAM do 6Kb, dodavanjem posebne pločica RAM bi mogao da se proširi do 54 kilobajta). ROM i RAM možemo da zamislimo kao skup ćelija koje su poređane u red i numerisane brojevima od &OOO do &FFFF. U svaku ćeliju može da se smesti bilo koji broj između &00 i &FFFF ili, da budemo precizniji, u nekim od ćelija već se nalaze fiksirani brojevi. Ove ćelije se nalaze na početku memorije (adrese &000 do &0FFF) i čine ROM — deo memorije u koji je upisan sistemski program koji omogućava ispravno funkcionisanje našeg računara. U ostale ćelije možemo slobodno da upisujemo sadržaj koji će biti interpretiran kao mašinski program koji može da se izvršava.

Adrese &1000 do &1FFF odgovaraju ROM-u koji se ne nalazi u vašem računaru — to je proširenje koje će jednoga dana biti na raspolaganju ako za njega bude interesa. Adrese &2000 do &27FF obuhvataju ćelije koje, u sadašnjoj situaciji, za nas nisu od interesa: računar ih interno koristi za rad sa tastaturom. Informacije sadržane u ćelijama &2800 do &2C39 su veoma značajne: tu se nalaze sistemske promenljive, video-memorija (o sistemskim promenljivima i video memoriji ćemo govoriti docnije), aritmetički akumulatori i stek. Počev od &2C3A u memoriju se smešta bejzik program (ukoliko sadržaj jedne sistemske promenljive nije promenjen, ali i o tome docnije) dok je kraj memorije rezervisan za nizove: od &37FF (ili &3FFF ako imate RAM od 6 kilobajta) se unazad prema bejzik programu, smešta alfanumerički niz X$(I), a ispod njega numerički niz A(I). Ukoliko se bejzik program i numerički niz „susretnu" na sredini memorije, računar ispisuje SORRY i odbija da dodaje nove elemente numeričkom nizu.

Koristeći naredbu BYTE možemo da proverimo i, po potrebi, promenimo sadržaj bilo kog bajta (ćelije) memorije. Otkucajmo, na primer, PRINT BYTE (&2BA8) i pritisnimo uobičajeno **RET** — na ekranu će se pojaviti broj 11. To znači da ćelija čija je adresa &2BA8 sadrži broj 11 (dekadno). Otkucajmo dalje BYTE &2BA8, 12 i primetimo da se slika na našem monitoru pomerila udesno. PRINT BYTE (32BA8) će nas uveriti da je prethodna naredba naložila računaru da promeni sadržaj ćelije &2BA8 i u nju smesti broj 12 umesto dosadašnjeg 11. Samo pomeranje slike nije direktno izazvano naredbom BYTE — ćelija &2BA8 kazuje računaru gde počinje slika, pa se promene njenog sadržaja trenutno primećuju. Pouka je da je vrlo opasno menjati sadržaje memorije pomoću BYTE ukoliko tačno ne znamo šta radimo — greška može da izazove brisanje programa pa čak i blokiranje računara posle koga ni RESET neće pomoći — moraćemo da isključimo i ponovo uključimo kompjuter, što je, kao što znamo, lek za sve probleme. U sadašnjem slučaju dovoljno je da otkucamo BYTE &2BA8, 11 i sve će se vratiti u normalno stanje.

Evo još jednog primera upotrebe naredbe BYTE: želimo, na primer da osvetlimo sve tačke na ekranu. Ako to radimo pomoću DOT, posao će potrajati prilično dugo. Ako koristimo PRINT, nećemo nikako moći da osvetlimo poslednji karakter. Na svu sreću, program

```
10 FOR I=&2800 TO &29FF
20 BYTE I, 255: NEXT I
```

Rešiće problem u prihvatljivom vremenu. Možemo da izmerimo vreme potrebno za ovu jednostavnu operaciju (ako ste dobro upoznati računar „galaksija", povećajte preciznost merenja koristeći računarev časovnik) obzirom da ćemo isti problem docnije rešiti i na mašinskom jeziku.

26. Nove mogućnosti rada sa alfanumericima — naredba PTR

Izdvajanje slova iz neke alfanumerička promenljive je do sada bilo skopčano sa velikim problemima (praktično nemoguće) što je onima koji su pisali programe moralo da prired-

poruku WHAT? i prekid rada programa). Smisao naredbe GOTO je sasvim jasan čak i za onoga ko nije imao nikakvih kontakta sa računarima: kada kompjuter naiđe na nju, preći će na izvršavanje linije broj 10. Na ovaj način smo sastavili program koji je „opasan": kada ga jednom startujemo on će se neprekidno izvršavati, stalno nas pitajući za podatke. Kada iscrpimo sve podatke, izvršavanje programa postaje besprędmetno i treba ga prekinuti. Jednostavno ćemo pritisnuti taster **BRK** i računar će ispisati BREAK 15 (što znači da je, u momentu kada smo ga zaustavili, izvršavao liniju broj 15) i uobičajeno READY. Zatim možemo da otkucamo NEW i tako obrišemo program, ali ćemo se za sada uzdržati od toga — program možemo da proširimo upoznajući dve nove naredbe.

Naredba GOTO se izvršava uvek na isti način — kada računar naiđe na nju, prelazi na izvršavanje naredbe čiji je broj naveden iza GOTO (možemo da napišemo i GOTO A ako je promenljivoj A ranije dodeljen broj linije na koju računar treba da ide, ali ćete ovo indirektno adresiranje, bar u početku, ređe koristiti). No, postoji naredba koja izaziva različito dejstvo u zavisnosti od nekih parametara — to je naredba IF. Iza IF se nalazi neka pogodbena relacija čiju tačnost računar ispituje. Da bismo ovo razumeli, dodaćemo u naš program naredbu **18 IF S=0 STOP**. Treba pre svega, da razumemo njenu sintaksu. STOP je naredba koju tek upoznajemo i koja jednostavno izaziva prekid rada. S=0 iza IF naredbe ne označava kao do sada da se promenljivoj S dodeljuje vrednost nula već navodi računar da ispita da li je vrednost promenljive S jednaka nuli. Ako jeste, izvršava se naredba STOP i računar prestaje sa radom, a ako nije, naredba STOP biva ignorisana i računar prelazi na izvršavanje sledeće naredbe.

Uz IF se često koristi i naredba ELSE. Možemo da unesemo novu naredbu 18 koja glasi

```
18 IF S=0 STOP: ELSE PRINT "NASTAVLJAM SA RADOM"
```

Ukoliko je S zaista nula, računar će se zaustaviti. Ako nije, računar će izvršiti naredbu iza ELSE, a zatim nastaviti da izvršava program. Dve tačke ispred ELSE za sada morate da primite „zdravo za gotovo" — njihova uloga će vam postati jasna nešto kasnije.

Kakav je smisao promena koje su izvršene? Korisniku je sada pružena mogućnost da, kada računar traži podatak o rastojanju gradova, otkuca 0 i pritisne **RET**. Računar će prestati sa radom i ispisati READY, eliminišući potrebu za pritiskanjem tastera **BRK**.

Umesto znaka = u pogodbenoj relaciji može da se nađe znak manje (na primer IF A<B PRINT "PRVI JE MANJI": ELSE PRINT "DRUGI JE MANJI ILI SU JEDNAKI") ili veće (na primer IF X>0 PRINT "BROJ JE POZITIVAN": ELSE PRINT „BROJ JE NEGATIVAN").

Ako eksperimentima sa IF naredbama poremetimo naš program koji će nam za dalji rad biti potreban, moraćemo da obrišemo suvišne linije. Brisanje linije broj 18, na primer, je vrlo jednostavno: otkucamo 18 i pritisnemo **RET**. Metoda je pogodna ali stvara jedan artefakt: želimo, na primer, da izračunamo koliko je 31/17 i zaboravimo da otkucamo PRINT. Računar će shvatiti da smo uneli programsku liniju broj 31 i obrisati eventualnu naredbu našeg programa koje ima isti broj, zamenivši je besmislenom linijom /17.

8. Prijavljivanje grešaka

Već smo upoznali poruke WHAT? i HOW? koje bivaju ispisane na ekranu kada našem kompjuteru, nešto nije jasno. Poruka WHAT? se uglavnom odnosi na sintaksne greške: upotrebili smo naredbu koju naš računar ne poznaje ili smo valjanu naredbu napisali na pogrešan način (PRINT AB, na primer, nema smisla jer se množenje ne podrazumeva). Poruku HOW? računar izdaje kada smo prekoračili opseg brojeva sa kojima se računa, kada smo naredbu GOTO (ili naredbu CALL koju tek treba da upoznamo) propratili linijskim brojem nepostojeće naredbe, kada smo pozvali previše potprograma i u sličnim situacijama.

Pored HOW? i WHAT?, postoji sistemska poruka sa kojom se nećemo sresti u skoroj budućnosti: SORRY. Ona nas obaveštava da je ono što smo želeli da učinimo sasvim moguće i ispravno napisano, ali da šest kilobajta (ili više ako smo proširili memoriju) nije dovoljno za smeštanje potrebnog programa ili podataka. Ako ovo uputstvo niste čitali „na preskok", upoznavši tako matrice A i X$, **jedini** način da na ekranu vidite

SORRY je da otkucata neki program koji ima više od pet kilobajta (računar „galaksija" u osnovnoj verziji ima 6 kilobajta RAM memorije, ali se nešto više od jednog kilobajta koristi za neka interne potrebe računara).

Ukoliko računar prijavi grešku u okviru neka komande (otkucali smo, na primer, RAN umesto RUN), na ekranu će se pojaviti jednostavno WHAT?, HOW? ili SORRY. No, ako računar nađe ne grešku u programu, on će, pored uobičajene poruke, ispisati čitavu liniju u kojoj je nastupila greška, stavljajući znak pitanja na ono mesto u liniji ne kome je naišao ne nerazumljiv simbol ili konstrukciju. Tako ne bi trebalo da bude teško da locirate grešku i, prekucavajući liniju ili koristeći komandu **EDIT**, dovedete program u ispravno stanje. U početku ćete verovatno mnogo grešiti kod oblika (sintakse) naredbi, ali će se docnije greške svesti ne pogrešno kucanje i, na žalost, logička greške (najopasnije su greške koje računar ne primećuje) — program se na prvi pogled korektno izvršava, ali su krajnji rezultati pogrešni.

9. Rad sa kasetofonom — SAVE, OLD i OLD?

Znanje koje smo stekli nam omogućava da pišemo relativno komplikovane programe. Međutim, svaki put kada isključimo računar ti programi bivaju obrisani pa ih, po ponovnom uključivanju, moramo kucati, što je veoma neprijatan posao. Na svu sreću, računar „galaksija" može da se poveže sa kasetofonom i snima programe na standardne kasete. Programi snimljeni na kasetu nemaju imena pa može da se desi da, nekoliko meseci po snimanju, zaboravimo šta koji program radi. Zato je korisno na početku programa dodati liniju **1! OVO JE PROGRAM ZA RAČUNANJE**

Uzvičnik na početku ove linije govori računaru da ona sadrži komentare koji su za njega nebitni: pri izvršavanju programa linije sa uzvičnicima bivaju ignorisane. Samo se po sebi razume da linija koja počinje uzvičnikom ne mora da bude prva u programu. Možemo, na primer, da pre svaka grupe naredbi ubacimo komentar koji objašnjava njihovo dejstvo, ali treba imati u vidu da ovakve naredbe troše jedan bajt memorije po svakom upotrebljenom slovu što je u uslovima ograničenog memorijskog prostora, često luksuz.

Kada pripremimo program za snimanje, jednostavno povezujemo odgovarajući džek sa priključkom MIC na kasetofonu, startujemo snimanje (dirke **PLAY** i **REC**), kucamo SAVE i pritiskamo **RET** (SAVE — sačuvaj). Ekran će se potpuno zamračiti i ovakvo stanje će potrajati izvesno vreme. Kada računar snimi program, ekran će oživeti sa novom porukom READY. Sačekajmo još nekoliko trenutaka i prekinimo snimanje.

Posle snimanja je vrlo korisno izvršiti verifikaciju. To što se na ekranu pojavilo READY još ne mora da znači da je program verno prenet na kasetu: moguće je da je traka bila oštećena (ovo oštećenje može da bude prisutno i kod potpuno nove kasete i ne mora biti uočljivo golim okom) ili da je kasetofon u nekom trenutku za momenat promenio brzinu. Ukoliko je nastupila neka greška ovog tipa, možemo da imamo velikih problema: isključimo računar i, posle pokušaja da učitamo program koji smo razvijali satima ili danima, utvrdimo da je on nepovratno izgubljen. Ako izvršimo verifikaciju, bićemo upozoreni na problem, pa ćemo moći da snimimo program na neku drugu kasetu.

Da bismo izvršili verifikaciju, moramo da premotamo kasetu na mesto koje prethodi početku snimljenog programa (veoma je preporučljivo posedovati kasetofon sa brojačem mada se i bez njega program prepoznaje po veoma karakterističnom zvuku, teško da ste ikada čuli neprijatnije zujanje) i zaustavimo ga. Povežimo utičnicu **EAR** na kasetofonu sa odgovarajućim priključkom računara, postavimo nivo zvuka blizu maksimuma i otkucajmo **OLD? RET.** Zatim startujemo kasetofon (taster **PLAY**) i čekamo. Može da se dogodi jedna od tri stvari.

Prva i najpovoljnija je da računar posle izvesnog vremena aktivira ekran i napiše READY. To je znak da je program korektno snimljen i uspešno verifikovan.

Sledeća mogućnost je da se, u toku učitavanja programa, ekran osvetli sa porukom WHAT? i uobičajenim READY. To znači da je računar upravo primetio neslaganje onoga što je u memoriji i onoga što je na kaseti. Možemo da pokušamo još jednu verifikaciju povećavajući nivo zvuka ili ga smanjujući, ali je vrlo verovatno da ćemo morati ponovo da snimamo program.

Koristeći upravo stečena znanja, logičko I ćemo lako zamenjivati množenjem, a logičko ILI — sabiranjem. Primer naveden tri pasusa ranije, na primer, može da se sažme u naredbu

```
10 IF (X>Y)*(Z>2) PRINT „USLOV JE ISPUNJEN" ELSE PRINT „USLOV NIJE ISPUNJEN"
```

23. Dalje mogućnosti naredbe HOME

Naredba HOME, kao što smo rekli, nalaže računara da obriše sadržaj ekrana i o njoj se, izgleda, ne mora više razgovarati. No iza HOME može da se nađe i neki broj između 1 i 512 koji unekoliko menja smisao ove naredbe.

Kada računar ispuni čitav ekran, prva linija nestaje, sve ostale se pomeraju za jedan red nagore, a novi tekst se pojavljuje u poslednjem (šesnaestom) redu ekrana. To je prirodan tok stvari, ali ponekad može da bude korisno da se jedan deo ekrana zaštiti od automatskog brisanja. Na tom delu ekrana može da se nađe neki tekst koji u toku čitavog rada programa treba da bude vidljiv (na primer, sažeto uputstvo za upotrebu kod neke igre). Da bismo, na primer, zaštitili prva tri reda ekrana, izvršićemo sledeći jednostavni program

```
10 HOME
20 PRINT „OVI REDOVI": PRINT „SU": PRINT „ZAŠTIĆENI"
30 HOME 96
```

Zatim kucajmo neke naredbe i, kada ostatak ekrana bude ispunjen, primetimo da je „pokretan" samo deo ispod trećeg reda.

Zaštita prvih redova nije potpuna: pomoću **PRINT AT** možemo da menjamo njihov sadržaj, pa čak i da ga obrišemo pritisnuvši **SHIFT DEL**. Međutim, zona na ekranu ostaje zaštićena čak i ako promenimo njen sadržaj — automatski scroll (pomeranje redova) je ne remeti.

Postoji jedno ograničenje vezano za zaštitu ekrana: kada je ona aktivirana, uobičajeni fini scroll (pomeranje redova nije naglo nego prijatno za oči) je isključen. To znači da će listanje programa biti oko tri puta brže nego obično, ali da će ga biti teže pratiti obzirom da redovi prebrzo „preleću" ekran. Ako želimo samo da isključimo fini scroll bez zaštite dela ekrana, otkucaćemo HOME 512. Ovo stanje, kao i zaštita bilo kog dela ekrana, opoziva se sa HOME.

24. Heksadecimalni sistem — oznaka &

Za rad sa mašinskim jezikom je pogodno koristiti heksadecimalni brojni sistem. U njemu se, uz cifre 0, 1, 2, 3, 4, 5, 6, 7, 8 i 9 koriste i „cifre" A, B, C, D, E i F. Tako, na primer, uobičajeni broj 10 se u heksadekadnom sistemu piše kao OA ili jednostavno A, 11 je OB, 12 — OC, 13 — OD, 14 — CE, 15 — OF, a 16 se piše kao 10. Ako u broju nema slova A, B, C, D, E i F, lako može da dođe do zabune u kom je sistemu pisan. Problem se u programiranju (a ponekad i u matematici) rešava tako što se ispred heksadekadnog broja stavlja **oznaka &** (na primer &10, &5A i slično). Ova oznaka je poznata i računaru „galaksija": otkucajmo, na primer, PRINT &000A (ili samo PRINT &A) i pritisnimo **RET** — na ekranu će se pojaviti broj 10 koji predstavlja prevod heksadekadnog broja OOOA u uobičajeni dekadni sistam.

Kada god u nekoj naredbi treba da napišemo neki dekadni broj, možemo da ga zamenimo heksadekadnim pri čemu ispred njega stavljamo znak &. Tako, na primer, možemo da zamenimo naredbu HOME 32 naredbom HOME &20, znajući da se u svakom redu nalazi po &20 (&20=32) karaktera. Treba voditi računa o činjenici da se oznakom & mogu dobiti i negativni brojevi: &8001 nije 32769 nego — 32767.

Rad sa heksadekadnim sistemom postaje posebno značajan tek sa naredbama BYTE i WORD.

Sa mašinskim programiranjem, ipak, ne možemo da počnemo baš odmah. Potrebno je da upoznemo neka tajne našeg računara i shvatimo kako je organizovana njegova memorija, kako možemo slobodno da menjamo njen sadržaj i kakvim se opasnostima pri tome izlažemo. Najpre ćemo pomenuti jednu karakteristiku koja može u mnogome da nam olakša manipulisanje sa računarom: mogućnosti skraćivanja naredbi.

Svaka naredba koju računar „galaksija" poznaje i koja ima više od dva slova može da se skrati. Umesto, na primer, da pišemo PRINT 10*2, možemo da otkucamo jednostavno P. 10*2. Tačka iza P govori računaru da je pred njim skraćena naredba PRINT. Ovako skraćena, naredba PRINT zauzima dva bajta memorije (PRINT zauzima pet), brže se izvršava i lakše kuca. Na žalost, skraćivanjem naredbi degradiramo izgled bejzik programa i činimo program znatno nerazumljivijim za početnika.

Bejzik kojim je opremljen računar „galaksija" je tako koncipiran da svaka naredba može da se skrati što eliminiše potrebu da navedemo tabelu skraćenih oblika. Ovo, na prvi pogled, izgleda nemoguće: već smo, na primer, upoznali naredbe INPUT i INT, NEW i NEXT i slične koje počinju istim slovom. Verovali ili ne, računar „galaksija" u ovoj prilici može ograničeno da čita vaše misli — skratite bilo koju naredbu na jedno slovo i on će pogoditi šta ste želeli da učinite! Objašnjenje je jednostavno: INPUT predstavlja naredbu koja počinje od početka neke linije bejzika ili od oznake :, a INT funkciju koja se nalazi samo desno od znaka jednakosti gde naredba INPUT svakako ne može da se nađe. Isto tako RETURN nema smisla u komandnom, a RUN u programskom modu, pa svaka od njih može da se zameni sa R. O svemu ovome ne morate mnogo da mislite: jednostavno skratite svaku naredbu koja ima više od dva slova na jedno slovo i stavite tačku, a računar se nikada neće zbuniti.

22. Računanje sa logičkim iskazima

Obzirom da se u IF naredbi mogu naći samo logički izkazi u kojima se koriste oznake <,> i=, svi oni koji su pisali iole složene programe susretali su se sa problemima tipa: potrebno mi je da računar ode na liniju broj 100 ako je X manje ili jednako Y. Ukoliko ste se dosetili trika koji ćemo da izložimo, zaslužujete sve komplimente i imate sve šanse da postanete dobar programer.

```
10 IF X>Y ELSE GOTO 100 (ili, s obzirom na prethodno poglavlje
10 IFX>YE G.100)
```

Šta se desilo? Ako je X veće ili jednako Y, odgovor na pitanje „da li je X manje od Y" je negativan, pa računar izvršava ono što se nalazi iza ELSE. Ukoliko je X manje od Y, ELSE biva ignorisano i računar prelazi na sledeću liniju.

Kako smo sli da izostavimo dve tačke ispred ELSE? To u ovom slučaju sme da se uradi obzirom da ne postoji nikakva dvoznačnost i da se tako štedi jedan bajt RAM-a. U radu sa računarom „galaksija" ćete primetiti da u mnogim slučajevima neke naredbe mogu da se iskoriste u pojednostavljenom obliku koji nije izložen da bi se izbegla konfuzija kod početnika. U našem primeru dve tačke ispred ELSE ne bi nikako zasmetale računaru!

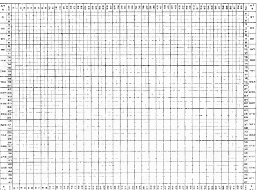
Ovaj trik je ipak nedovoljan: često želimo da se neka naredba izvrši ako je ieno nekoliko logičkih iskaza (ako je, na primer, X veće od Y i Z veće od 2). Računar „galaksija" omogućava i korišćenje logičkih operacija i odnosno ILI, ali na donekle indirektan način.

Svakom logičkom iskazu koji je tačan dodeljuje se vrednost jedan, a onome koji je netačen — nula. Otkucajmo, na primer, PRINT 2=2 i na ekranu će se pojaviti broj jedan obzirom da je dva jednako dva. Možemo, isto tako, da napišemo PRINT (B=2)+(A=3). Ukoliko je B zaista dva a A tri, na ekranu će se pojaviti broj 2 (1+1=2). Ukoliko je B jednako dva ali je A različito od tri ili ako je B različito od dva a A jednako tri, na ekranu će se pojaviti broj 1 (1+0=0+1=1). Ako je, najzad, A različito od tri, a B od dva, na ekranu će se pojaviti nula (0+0=0).

Naredba IF izračunava vrednost izraza koji je naveden iza nje i smatra da je on netačan ako je jednak nuli, a tačan ako je različit. U tom smislu je IF S=O ELSE GOTO 100 potpuno ekvivalentno sa IF S GOTO 100, pri čemu je druga naredba kraća i brže se izvršava.

Treća i najneprijatnije mogućnost je da se ne dogodi ništa — ekran je neprekidno crn iako je prošlo daleko više vremena nego što je trajalo snimanje programa (ili, ako imamo kasetofon sa brojačem, pređeno je mesto na kome se nalazi kraj programa). To znači da je nivo zvuka bio preslab pa računar nije pronašao početak programa ili da usled neka pogrešne manipulacije sa utikačima, ništa nije ni snimljeno na traku. Pritisnimo dugme RESET koje nalazi sa zadnje strane računara i primetimo (kucanjem naredbe LIST) da je naš program i dalje u memoriji. Možemo da pokušamo novu verifikaciju ili, po potrebi, novo snimanje.

Treća opcija pri radu sa kasetofonom je unošenje programa sa traka i verovatno smo je savladali pri korišćenju demo-kaseta. Povežimo računar sa kasetofonom na način opisan kod opcije OLD?, otkucajmo OLD **RET** i startujmo traku (taster PLAY). Posle izvesnog vremena, program će biti učitan i na ekranu će pisati READY. Ukoliko je to jedini prisutni komentar, program je vrlo verovatno ispravno upisan i možemo da ga startujemo sa RUN. No, ako se pre READY našlo WHAT?, računar je primetio grešku pri upisu. Možemo da otkucamo LIST (ne zaboravimo da listanje programa traje samo dok držimo pritisnut taster **RET**) i pogledamo na šta program liči: sasvim je moguće da su grupe slova promenjene ili ispuštene. Najbolje je da u ovakvom slučaju pokušamo upisivanje još jedanput, menjajući jačinu tone u nadi da će sve biti u redu. Ukoliko ne uspemo, ostaje nam da pokušamo da ispravimo program što može da bude vrlo neprijatno, pogotovu ako nismo njegov autor. Iz iskustva možemo da kažemo da se problemi ovoga tipa ne javljaju kod verifikovanih programa, osim kada su snimani na jednom a učitavani sa drugog kasetofona. No, ako nam je neki program važan, želećemo da eliminišemo sve moguće izvore probleme: u tom slučaju je korisno da snimimo program dva puta i to po mogućnosti na različite kaseta. Obzirom da snimanje oduzima svega 2 minuta po kilobajtu, ova mera predostrožnosti neće predstavljati veliki gubitak vremena.



10. Dalje mogućnosti naredbe PRINT, PRINT AT, HOME

Na samom početku smo rekli da će ovo uputstvo biti organizovano „po krugovima": prvi krug predstavlja upoznavanje tastature i rad sa demo-kasetom, drugi osnova programiranja a treći, u koji upravo ulazimo, „luksuzni" naredbe i njihove verzije. Polazimo, naravno, ponovo od PRINT.

Do sade smo iza PRINT stavljali tekst pod znacima navoda ili neki brojni izraz koji je izračunavan i prikazivan na ekranu. Na taj način smo u jednom redu mogli da ispišemo samo jednu rečenicu što je, ako ništa drugo, nestatski. U okviru jedne naredbe PRINT može da se nađe više promenljivih, izraza i komentara koji će biti prikazani u istom redu. Računar, međutim, zahteva da ovi komentari budu odvojeni jedan od drugog znakom ; ili običnim zarezom. Ukoliko su dve stavke u naredbi PRINT odvojene simbolom ; računar će ih nadovezati jednu na drugu, a ako je upotrebljen zarez — razmaknuće ih tako da druga počinje od osmog, šesnaestog ili dvadeset četvrtog mesta u redu. Ovu naredbu ćemo najbolje shvatiti na primeru.

Vratimo se na program za izračunavanje vremena putovanje između dva mesta i obrišimo linije 40, 50, 70 i 80 a zatim otkucajmo

70 PRINT INT(T);"ČASOVA";M;"MINUTA"

Izvršimo program i unesimo uobičajene podatke za rastojanje Beograd—Dubrovnik i brzinu 800 km/h. Na ekranu će se pojaviti izveštaj

0ČASOVA 27MINUTA

Nije baš sjajno — redovi više nisu odvojeni ali je računar očigledno preterao sa zbijanjem. Ispred svakog broja je, doduše, ostavljao po jedno prazno mesto (to mesto u stvari nije prazno — na njega bi bio smešten znak minus da je broj bio negativan) ali nam je neophodan blanko simbol i iza njega. Jedan od načina da rešimo problem je da otkucamo

70 PRINT INT(T),"ČASOVA",M,"MINUTA"

ali će tako rečenica biti nelogično mnogo izdeljena na polja. Pravo rešenje je naredba

70 PRINT INT(T);" ČASOVA ";M;" MINUTA"

Upoznavanjem simbola , i ; još nismo iscrpli sve mogućnosti naredbe PRINT: ona može da posluži i za pisanje teksta na proizvoljnom delu ekrana. Da bi ovakvo pisanje imalo nekog smisla treba najpre da upoznamo naredbu za brisanje čitavog sadržaja ekrana. Ovaj efekat se u komandnom modu postiže istovremenim pritiskom na tastere [SHIFT] i [DEL], a u programskom — izvršavanjem naredbe HOME (HOME nalaže računaru da „vrati kurzor kući" koja se nalazi u levom gornjem uglu ekrana. Pri tom se čitav sadržaj ekrana briše).

Na slici je prikazana mapa ekrana. Kao što se vidi, on je podeljen na 16 redova, a svaki od tih redova na 32 mesta. Mesta su numerisana brojevima od 0 (karakter u levom gornjem uglu ekrana) do 511 (karakter u desnom donjem uglu). Naredba PRINT AT može da počne da štampa od proizvoljnog mesta koje je navedeno iza AT; pun oblik naredbe je dakle PRINT AT 32, "RAČUNAR GALAKSIJA" — u ovom primeru tekst „računar galaksija" se pojavljuje na početku drugog reda dok sadržaj prvog ostaje nepromenjen.

Pri prikazivanju nekih komplikovanijih tekstova treba posebno paziti na jednu stvar: ako najpre ispisujemo tekst u desnoj polovini neke linije a zatim u levoj, neka PRINT AT naredba može lako da poremeti postojeći sadržaj nekog reda. Da se obezbedimo od toga završavaćemo svaku naredbu PRINT znakom ; (pišemo na primer, PRINT AT 300, "RAČUNAR";)

broj i ne pritisne RET, što znači da je vreme za razmišljanje neograničeno. Osim toga naredba INPUT ispisuje na ekranu znake koje korisnik kuca i tako kvari sliku koja je na njemu prikazana. Zato računar „galaksija" poseduje naredbu KEY.

NAREDBA KEY (N)

TASTER	N	TASTER	N	TASTER	N	TASTER	N	TASTER	N	TASTER	N	TASTER	N
A	1	I	9	Q	17	Y	25	1	33	9	41		
B	2	J	10	R	18	Z	26	2	34	;	42	RPT	50
C	3	K	11	S	19	↑	27	3	35	:	43		
D	4	L	12	T	20	↓	28	4	36	,	44	LIST	52
E	5	M	13	U	21	←	29	5	37	=	45	SHIFT	53
F	6	N	14	V	22	→	30	6	38	.	46		
G	7	O	15	W	23	BLANKO	31	7	39	/	47		
H	8	P	16	X	24	Ø	32	8	40	ENTER	48		

Iza KEY se, u zagradi, nalazi broj koji se određuje iz tablica koja je data na slici. Vidi se da svakom tasteru odgovara neki broj, pa postaje jasno za šta KEY (n) služi: ova naredba omogućava ispitivanje da li je neki taster pritisnut ili nije. Posmatrajmo, na primer, program

10 IF KEY (1) PRINT "PRITISNUO SI A" ELSE GOTO 10

Program se neprekidno izvršava sve dok korisnik ne pritisne taster A, zatim se na ekranu pojavljuje odgovarajući komentar posle čega računar prestaje sa radom.

Na demo kaseti je naredba KEY izuzetno mnogoeksploatisana, naročito za ispitivanje pritisnutosti neke od strelica (u okviru naredbe INPUT čak se i ne može uneti strelica kao karakter!) pa će analiza programa sa nje pružiti neiskusnom korisniku predstavu o značaju ove naredbe.

KEY(0) ima funkciju koja je u mnogo čemu različita od funkcije do sada opisane naredbe KEY. KEY (0), naime, daje kod znaka koji je pridružen tasteru koji korisnik pritisne (kod se računa prema tabeli koja je data u diskusiji naredbe CHR$). Naredba IF KEY(0) se, dakle, neće pravilno izvršavati, ali zato može da se napiše A=KEY(0). Kada računar u toku izvršavanja programa naiđe na ovu naredbu, on će čekati sve dok korisnik ne pritisne neki taster, a zatim će vrednost kôda (broj 80 ako korisnik pritisne taster P, na primer) znaka koji je ispisan na tom tasteru pridružiti promenljivoj A.

10 HOME
20 PRINT CHR$(KEY(0)); GOTO 20, na primer, prenosi na ekran slova koje korisnik kuca pa se tako, na neki način, imitira „normalan" rad računara. Međutim, program
10 X$=""
20 X$=X$+CHR$(KEY(0))
30 GOTO 20 na izvestan način imitira naredbu INPUT X$, s tim što se uneseni tekst smešta u X$ bez kvarenja izgleda ekrane.

21. Skraćivanje naredbi

Ovim poglavljem započinjemo poslednji krug: to je deo uputstva za upotrebu koji je namenjen onima koji su solidno savladali osnove programiranja i upoznali sve do sade izložene karakteristike računara „galaksija". Njim započinje deo koji će, silom prilike biti manje jasan i zahtevati više napore za savladavanje. Za to postoje jaki razlozi: programiranje na mašinskom jeziku, koje želimo da upoznemo, predstavlja viši stupanj, neku vrstu nadgradnje bejzika i nije namenjeno neiskusnim korisnicima. Ovim, naravno, ne želimo nikoga da plašimo: mašinsko programiranje nije teško, ono zahteva samo malo više rade i nema smisla upuštati se u njega dok nismo savladali bejzik.

```
10 HOME Y$=„10 15 20 15" (ili bilo koje drugo vreme)
20 DOT*
30 PRINT AT 20,Y$
40 GOTO 30
```

Ovako rešen časovnik ima tri mane: prva je što se isključivanjem računara briše tekuće vreme, druga što časovnik ne radi dok se program snima ili učitava sa kasete, a treća što, ako je u programu korišćena naredba DOT*, moramo da pazimo da li su dve tačke treći simbol promenljive Y$. Ako jesu, računar će smatrati da Y$ sadrži tekuće vreme, pa će menjati njen sadržaj što može da proizvede neželjene posledice.

19. Alternativni metod primanja podataka — TAKE i

Do sada smo primali podatke isključivo pomoću naredbe INPUT, a dodeljivali sadržaje promenljivima i pomoću komandi tipa A=123. Često nam je, međutim, u početku programa neophodno da dodelimo vrednosti mnogobrojnim promenljivima i elementima matrice A. U takvim slučajevima naredba TAKE može da bude neobično praktična.

Iza TAKE se navodi lista promenljivih kojima treba dodeliti vrednosti. To mogu da budu numeričke promenljive, elementi matrice A, alfanumeričke promenljive ili elementi niza X$ (ako je ovaj prethodno definisan sa ARR$). Elementi ove liste se odvajaju zarezima. U početku programa će se, dakle, često naći naredba

```
10 TAKE A,B,C,X,Z
```

Ali, odakle računar može da uzme (TAKE=uzmi) ove podatke? U okviru programa mora na proizvoljnom mestu (pre ili posle naredbe TAKE, po želji) da se nađe jedna ili više linija koje počinju simbolom #. Ovaj simbol obaveštava računar da se u linijama koje ga sadrže nalazi tablica podataka koji će biti pročitani sa TAKE. Brojevi i reči u ovim tabelama treba da budu razdvojeni zarezima.

Ako se, na primer, u programu nalazi naredba

```
100#10, 20, 30, ... 11, 79, 779,
```

a računar izvrši ranije navedenu TAKE naredbu, promenljiva A će dobiti vrednost 10, promenljiva B dvadeset itd., promenljiva Z će dobiti vrednost 79. Računar će „zapamtiti" dokle je došao sa čitanjem podataka pa će, ako naiđe na naredbu 50 TAKE L, promenljivoj L dodeliti broj 779.

Ako se u listi iza # nalazi manje podataka nego što se u TAKE naredbi traži, računar će prijaviti grešku. Obrnuto, ukoliko se ne učitaju svi podaci iz liste, računar neće reagovati, što je i prirodno.

Ponekad može da bude potrebno da se u okviru programa ponovo uzimaju isti podaci (potrebne su, na primer, početne vrednosti za neku igru, a želimo da se, po završetku svake partije, automatski započne nova sa istim početnim parametrima). U tom slučaju pre prve naredbe TAKE koja se nalazi u programu treba da stavimo TAKE 0 (nula a ne slovo o). Na taj način će računar započeti uzimanje podataka od samog početka prve # liste. Umesto nule, iza TAKE možemo da stavimo neki drugi broj. U tom slučaju računar pronalazi liniju sa tim brojem i priprema se da u sledećoj TAKE naredbi uzme podatke iz # liste koja se nalazi u toj liniji. Ukoliko u navedenoj naredbi nema # liste, računar će pronaći prvu sledeću takvu listu, odakle vidimo da je TAKE 0 samo specijalni slučaj naredbe TAKE n. Naredba RUN automatski izvršava TAKE 0.

20. Naredba KEY

U nekim igrama su potrebne brze reakcije igrača i naredba INPUT postaje sasvim nepogodna. INPUT A, na primer, prekida rad programa sve dok korisnik ne unese neki

11. Crtanje po ekranu — DOT, UNDOT

Prethodno poglavlje nije razjasnilo sve oznake sa mape ekrana: tamo su redovi podeljeni na po tri dela, a kolone na po dva. Na ovaj način su dobijene takozvane tačke ili „pikseli", kojih ima po 64 u svakom od 48 redova. Svaku ovu tačku (to je, zapravo, kvadratić) možemo slobodno da osvetlimo ili zatamnimo koristeći naredbe DOT i UNDOT (DOT znači tačka). Treba odmah da uočimo razliku između „tački" i karaktera: tačka je osvetljena ili zatamnjena, dok su karakteri daleko raznovrsniji (sva slova, brojevi, interpunkcijski znaci, prazno mesto ...); karakteri su adresirani brojevima od 0 do 511, a tačke na mnogo komforniji način — za obraćanje svakoj navodimo red u kome se ona nalazi i njen položaj u tom redu, najzad, tačke su šest puta manje od karaktera, pa njima mogu da se dobijaju bolje slike na ekranu.

Naredbu DOT prate koordinate tačke koju treba osvetliti (DOT 0,0, na primer, osvetljava tačku u levom gornjem uglu, DOT 1,0 tačku desno od nje, DOT 31,23 tačku u sredini ekrana, a DOT 63,47 tačku u desnom donjem uglu). Naredba UNDOT ima obrnuto značenje: ona zatamnjuje tačku čije su koordinate navedene na isti način. Program

```
10 HOME
20 DOT 31,23
30 DOT 40,40
40 UNDOT 31,23
50 UNDOT 40,40
60 GOTO 20
```

neprekidno uključuje i isključuje tačke čije su koordinate (31,23) i (40,40).

Naredba DOT može da se upotrebi i unutar IF, ali tada ima sasvim drukčiji smisao: linija 100 IF DOT 10,10 GOTO 1000 ne osvetljava tačku 10,10 nego ispituje da li je ta tačka osvetljena i, ako jeste, prelazi na liniju broj 1000.

Ukoliko je neka tačka osvetljena, primena naredbe DOT na nju neće promeniti ovo stanje; slično tome, UNDOT neće promeniti status tačke koja je već zatamljena. Naredbe DOT i UNDOT su veoma značajne i zaslužuju da ih upoznamo na dodatnim primerima. Ipak, sačekaćemo izvesno vreme obzirom da je za ovakvu demonstraciju neophodno realizovati cikluse (petlje), a to ćemo biti u stanju da uradimo daleko bolje kada upoznamo grupu novih naredbi.

12. Ciklusi — FOR-TO-NEXT-STEP

Često je potrebno da se određeni skup instrukcija ponovi određen broj puta i da se zatim nastavi sa izvršavanjem programa. Pretpostavimo da nam je potreban program koji izračunava faktorijel nekog broja. Faktorijel se definiše kao proizvod svih prirodnih brojeva zaključno sa brojem čiji se faktorijel traži (faktorijel broja 6 je, na primer, 1·2·3·4·5·6=720). Potrebno je, dakle, da počnemo od broja 1 i množimo ga redom sa 2, 3, 4, ... sve do nekog n. Ovaj problem može primenom naredbe IF da se reši na sledeći način:

```
10 INPUT N
20 Y=1
30 I=1
40 I=I+1
50 Y=Y*I
60 IF I<N GOTO 40 ELSE PRINT Y
```

Vidimo da je bilo neophodno uvesti naredbu koja inicijalizuje brojač (I=1), naredbu za povećavanje tog brojača kao i naredbu kojom ispitujemo da li je posao završen. Ovakvo rešenje, naravno, može da se primeni i u mnogo složenijim slučajevima, ali su takvi programi teži za sastavljanje (da smo, na primer, promenili redosled naredbi 40 i 50 u IF komandi bi morali da poredimo I sa N+1), manje pregledni i sporije rade. Naredba FOR

— TO i odgovarajuća naredba NEXT su pripremljene tako da olakšaju realizaciju ciklusa. Pre nego što pređemo na njihov detaljan opis, videćemo kako se prethodni problem rešava njihovom primenom

```
10 INPUT N
20 Y=1
30 FOR I=1 TO N
40 Y=Y*I
50 NEXT I
60 PRINT Y
```

Dobijeni program ima šest naredbi kao i prethodni ali u celini gledano zauzima manje memorije i brže se izvršava. Čak i ako vam je ovo prvi susret sa naredbama FOR — TO — NEXT verovatno ste shvatili da se ciklus odvija između naredbi 30 i 50. Računar po nailasku na naredbu 30 „inicira brojač" odnosno dodeljuje promenljivoj I sadržaj 1 (da je naredba 30 glasila FOR I=2 TO N promenljivoj bi bio dodeljen sadržaj 2 i tome slično) i „pamti" mesto na kome je ciklus počeo. Izvršavanje programa se nastavlja sasvim normalno sve dok računar ne naiđe na naredbu NEXT I („sledeće I"). U tom trenutku on dodaje jedan promenljivoj I i ispituje da li se došlo do N. Ako nije, ciklus automatski počinje od početka. Ako jeste, naredba NEXT I se preskače i računar prelazi na izvršavanje naredbe 60.

U okviru naredbe FOR može da se nalazi i opcija STEP (korak). Da smo umesto naredbe 30 otkucali FOR I=1 TO N STEP 2 računar bi svaki put kada naiđe na naredbu NEXT I uvećavao sadržaj promenljive I za dva pa bi računao proizvod Y = 1 3 5 7 ... N (za slučaj da je N neparan broj ovo bi bila funkcija bez posebnog imena koja se obeležava sa N!!, a ako je N paran rezultat ne bi imao matematičko značenje. Da bi program računao proizvod 2 4 6 8 ... N = N!! (za parne brojeve) naredbu 30 bi trebalo izmeniti tako da glasi FOR I=2TO N STEP 2)

Nema nikakve prepreke da STEP bude negativan. Program

```
10 FOR I=30 TO 0 STEP —1
20 PRINT I
30 NEXT I
```

će brojati od 30 do 0 i to unazad. Ipak postoji jedno ograničenje vezano za naredbe FOR—NEXT—STEP koje treba imati u vidu ako u praksi verovatno neće izazvati nikakve probleme: konstante ili promenljive koje figurišu u ovim naredbama moraju da imaju vrednosti celih brojeva; ako nemaju, računar će sam „odseći" decimale. Naredba FOR I=0.5 TO 11.5 STEP 0.5 će se izvršavati nekorektno obzirom da će računar podrazumevati STEP 0.

13. Slučajni brojevi — RND

Svi rezultati koje računar daje su precizne funkcije ulaznih veličina. Računar „galaksija" je opremljen jednom naredbom koja na prvi pogled odstupa od ove definicije: ona je namenjena generisanju slučajnih brojeva. Slučajni brojevi koje računar generiše su u stvari „pseudoslučajni" — rezultat su komplikovanih operacija nad sadržajima nekih sistemskih promenljivih i korisnik nema nikakvog načina da ih predvidi; za njega su dakle oni slučajni u dovoljnoj meri.

Otkucajmo NEW i unesimo jednostavni program

```
10 PRINT RND
20 GOTO 10
```

i startujmo ga sa RUN. Na ekranu će velikom brzinom početi da se pojavljuju brojevi koje istini za volju nećemo stići ni da pročitamo. Došao je dakle pravi trenutak da otkrijemo poslednju tajnu tastature našeg računara. Pritisnimo taster DEL i držimo ga pritisnutog. Događaji na ekranu su sa najednom „zamrzli" — izgleda da je izvršavanje programa zaustavljeno. Prvi utisak ovde ne vara: izvršavanje programa je zaista zaustavljeno i taj prekid će trajati sve dok je taster DEL pritisnut; njegovim otpuštanjem program nastavlja da se izvršava. Treba odmah uočiti razliku između **DEL** i **BRK**: po

koje se nikako drukčije nije mogla dobiti obzirom da znaci navoda ne mogu da budu deo nekog teksta pošto označavaju njegov početak i kraj.

U okviru naredbe X$= ... može da se nalazi promenljiva Y$ pa čak i sama promenljiva X$ ali uz jedno ograničenje: ona mora da bude sasvim na početku izraza. Dakle, naredba X$=X$+Y$ će nadovezati sadržaj promenljive Y$ na promenljivu X$ ali će X$=Y$+X$ izazvati neželjeno ponašanje (probajte!). Slično može da se napiše Y$=Y$+CHR$(60)+X$ ali ne i Y$=CHR$(99)+Y$+X$

Ako nam dve alfanumeričke promenljive nisu dovoljne, možemo da definišemo alfanumerički niz koji se zove X$(I) (ne treba ga ponovo mešati sa promenljivom X$). Na početku programa treba da stavimo naredbu ARR$(n) gde je n broj elemenata niza X$(I) koji nam je potreban (svaki od tih elemenata može da se upotrebi za čuvanje 16 znakova), a zatim možemo da postavljamo i menjamo sadržaje nekih od tih elemenata koristeći ranije izložena pravila za sabiranje (nadovezivanje) alfanumerika i funkciju CHR$.

Ponekad može da nam zatreba da uporedimo neka dva alfanumerika da bi saznali da li su oni jednaki. Alfanumerici se smatraju jednakim ako sadrže iste znakove na istim mestima i ako su jednake dužine. U tom smislu je „ABC"=„ABC" ali nije „ABC "=„ABC". Za poređenje alfanumerika koristimo naredbu EQ.

Možemo, na primer, da napišemo

```
100 IF EQ Y$,X$(10) GOTO 1000 ELSE PRINT „RAZLIČITI SU"
```

i računar će, ako su sadržaji promenljivih X$(10) i Y$ jednaki, preći na naredbu 1000, a ako nisu prikazati na ekranu RAZLIČITI SU i nastaviti sa radom. Možemo, jasno, da poredimo promenljive X$ i Y$, bilo koja dva elementa alfanumeričkog niza X$(I) ili bilo koju kombinaciju.

Za svakog početnika je prilično teško da pravi razliku između numeričkih i alfanumeričkih promenljivih. Naredba X$=„123" je, na primer, sasvim različita od naredbe X=123. U prvom slučaju smo dobili niz znakova koji ne podleže aritmetičkoj obradi (ako izuzmemo pomenuto nadovezivanje), a u drugom operativni broj 123. Promenljive X$ i X su sasvim različite i računar ih čuva na raznim mestima u memoriji.

U nekim slučajevima može da bude korisno da se neki alfanumerik koji se sastoji od cifara pretvori u broj. Za to nam služi naredba VAL. Otkucajmo jtavan program jednostavan program

```
10 INPUT X$
20 PRINT VAL(PTR X$)+1
```

i startujmo ga sa RUN. Kada računar zatraži od nas da otkucamo X$, odgovorimo sa 25 (posle toga naravno pritiskamo RET). Trenutak docnije na ekranu se pojavio broj 26. Računar je, naime, našao na naredbu PRINT, našao VAL (vrednost) alfanumeričke promenljive X$ i dodao joj jedan, dobivši rezultat 26. Napominjemo da sintaksa naredbe VAL koja je ovde korišćena ne može da bude jasna čitaocu koji nije čitao „na preskok" jer se u njoj koristi naredba PTR koju još nismo obradili. Za sada je, ipak, primite onakvu kakva jeste, a docnije će sve postati jasnije.

Kada unosimo X$ ne moramo da se ograničimo samo na brojeve: možemo kucamo čitav izraz, koristeći imena promenljivih i pazeći da on nema više od 16 simbola. Naredba VAL (PTR X$) će izračunati vrednost ovog izraza i pretvoriti ga u broj.

18. Časovnik — DOT *, UNDOT *

Računar „galaksija" je opremljen vrlo preciznim kvarcnim časovnikom koji će, po svoj prilici, naći upotrebu kao štoperica. Ovaj casovnik može, ako mu to naredimo, da prikazuje samo sate i minute ali može da bude i toliko precizan da prikazuje i pedesetinke sekunde. Za rad sa njim je potrebna promenljiva Y$ i naredbe DOT* i UNDOT*.

Najpre treba da postavimo tekuće vreme. Napišemo (u komandnom modu ili u programu) Y$=„21 45 20 50" i pritisnimo RET (ako želimo da časovnik pokazuje samo sate i minute napisaćemo Y$=„21 45", a ako želimo sate, minute i sekunde Y$=„21 45 20"; ukoliko želimo da merimo vreme napisaćemo Y$=„00 00 00 00"). Računar će smatrati da je sada tačno 21 čas 45 minuta, 20 sekundi i 50 stotinki, ali njegov časovnik još neće započeti da radi: treba da ga startujemo sa DOT*. Ako poželimo da zaustavimo časovnik, upotrebićemo naredbu UNDOT*. Evo, na primer, programa koji u desnom gornjem uglu ekrana neprekidno prikazuje tekuće vreme

17. Rad sa alfanumericima — CHR$, EQ, VAL, PTR ARR$

Do sada smo radili isključivo sa brojevima, dok smo komentare pisane standardnim srpskohrvatskim jezikom umetali isključivo između navodnika u PRINT naredbama i iza znaka! Računar „galaksija", međutim, poseduje i mogućnost rada sa nizovima znakova (pod znakom ovde podrazumevamo slovo, broj, interpunkcijski znak i prazan prostor) ili, kratko rečeno, alfanumericima.

Poput brojeva, za smeštanje reči koristimo promenljive koje se nazivaju alfanumerič-kim. U programu se prepoznaju po tome što se iza imena svake nalazi oznaka za dolar ($). U svaku alfanumeričku promenljivu može da se smesti tekst koji ima najviše 16 znakova, računajući i razmake, interpunkcijske znakove i slično.

Postoje samo dve fiksne anumeričke promenljive koje se nazivaju X$ i Y$. Njima možemo da dodeljujemo vrednosti korišćenjem standardnih bejzik naredbi, pri čemu ne smemo da zaboravimo da sadržaj uvek (osim kod izvršavanja naredbe INPUT) stavljamo između znakova navoda. Možemo, na primer, da napišemo X$=„ALFANUMERIK", a zatim PRINT X$ na ekranu će da se pojavi reč alfanumerik.

Pored uobičajenog prenošenja sadržaja jedne promenljive u drugu (naredba X$=Y$) raspolažemo i funkcijom CHR$. Ovu funkciju prati argument u zagradi koji je jednostavno neki broj između 0 i 255. Ovaj broj dobijamo iz tablice koja je data na sledećoj slici. Tablicu sličnu ovoj možete da dobijete na ekranu i ako izvršite sledeći jednostavni BASIC program:

TABLICA ASCII KARAKTERA

32		43	+	54	6	65	A	76	L	87	W
33	!	44	,	55	7	66	B	77	M	88	X
34	"	45	-	56	8	67	C	78	N	89	Y
35	#	46	.	57	9	68	D	79	O	90	Z
36	$	47	/	58	:	69	E	80	P	91	Č
37	%	48	0	59	;	70	F	81	Q	92	Ć
38	&	49	1	60	<	71	G	82	R	93	Ž
39	spec	50	2	61	=	72	H	83	S	94	Š
40	(	51	3	62	>	73	I	84	T	95	_
41	)	52	4	63	?	74	J	85	U		
42	*	53	5	64	spec	75	K	86	V		

```
10 FOR I=32 TO 255
20 PRINT I," ",CHR$(I),
30 NEXT I
```

Pogledajmo kako to izgleda na primeru. Naredba PRINT SHR$ (65) će, na primer, pokazati slovo A na ekranu. Funkciju CHR$ koristimo pri formiranju složenih alfanumeri-ka koje se obavija sabiranjem.

Alfanumerici jasno, nisu brojevi koje ima smisla sabirati. No, iako se za sabiranje alfanumerika koriisti isti znak kao i za sabiranje brojeva (plus), pojam ima sasvim drukčiji smisao: alfanumerici se nadovezuju. Ako, na primer, napišemo X$=„GA-LA"+CHR$(75)+„SIJA" a zatim PRINT X$ na ekranu će se pojaviti reč GALAKSIJA obzirom da CHR$(75) odgovara slovu K.

Na prvi pogled bi se reklo da naredbe poput gornje nemaju nekog velikog smisla: zar nije jednostavnije napisati X$=„GALAKSIJA"? No, posmatrajmo naredbu X$=CHR$(34)+„GALAKSIJA"+CHR$(34). Po njenom izvršavanju PRINT X$ daje reč „GALAKSIJA"

pritisku na BRK izvršavanje programa je trajno prekinuto i na ekranu piše READY, dok pritisak na DEL samo zaustavlja rad računara koji se nastavlja po otpuštanju tastera.

Kada znamo kako da prekinemo rad programa, možemo da se usresredimo na naredbu RND. Na ekranu se nalaze brojevi koje mi unapred ne možemo da predvidimo — oni su slučajan izbor vašeg računara. To su, u svakom slučaju, brojevi između 0 i 1 koji imaju po pet-šest decimala (ne treba se zbuniti ako neki od njih bude u eksponencijalnoj notaciji).

U praksi nam najčešće nisu potrebni decimalni brojevi između 0 i 1, nego celi brojevi između nekih a i b (ako, na primer, hoćemo da simuliramo kockicu za igru „čoveče, ne ljuti se" trebaju nam slučajno izabrani brojevi od 1 do 6). Prava je prilika da se podsetimo naredbe INT i naučimo formulu R=INT(RND*(B−A)+A). Njenom primenom računar generiše cele slučajne brojeve između A i B (ovi brojevi mogu da budu jednaki A ali ne mogu da budu jednaki B; matematičari bi rekli da se ovi brojevi nalaze u intervalu [A,B)).

Sastavimo program koji bi demonstrirao naredbe DOT i UNDOT. Najpre ćemo da obrišemo ekran, a zatim osvetljavati slučajne tačke. Ukoliko naiđemo na tačku koja je već osvetljena, zatamnićemo je.

```
10 HOME
20 X=INT (RND*64)
30 Y=INT (RND*48)
40 IF DOT X, Y UNDOT X,Y ELSE DOT X,Y
50 GOTO 20
```

Ako imate problema sa razumevanjem programa, moraćete da se vratite na poglavlje „crtanje po ekranu" i „naredba IF". Ako ga razumete, pokušajte da ga dopunite tako da se u desnom gornjem uglu ekrana neprekidno ispisuje broj osvetljenih tačaka. Ukoliko je generator slučajnih brojeva dobar, posle izvesnog vremena bi trebalo da se uspostavi „dinamička ravnoteža" — stanje u kome se približno jednak broj tačaka osvetljava i zatamnjuje u nekom intervalu.

14. Potprogrami — CALL i RETURN

U mnogim programima se javljaju grupe instrukcija koje se ponavljaju. Računar „galaksija", na primer, nije opremljen logaritamskom funkcijom, pa će mnogi poželeti da je dodaju. Oni će, dakle, koristeći neku numeričku formulu ili iterativni algoritam sastaviti program koji izračunava log X za zadato X i poželeti da ga koriste u mnogim prilikama. Ako, međutim, navode čitav skup instrukcija svaki put kada im logaritam zatreba, neće daleko stići: izdaće ih strpljenje ili memorija računara. Zato se ovakav skup instrukcija odvaja u poseban program koji se iz glavnog programa poziva kao potprogram.

Potprogram se ne započinje ni na kakav poseban način — bitno je samo zapamtiti broj njegove prve linije. Na kraju potprograma mora da se nađe naredba RETURN (=vrati se) koja nalaže računaru da nastavi sa izvršavanjem glavnog programa.

U glavnom programu se, kada je god potprogram potreban, piše naredba CALL koju prati broj prve linije potprograma (npr. CALL 1000). Sve će biti mnogo jasnije kada pogledamo primer. Sastavićemo program koji pravom linijom spaja tačke čije su koordinate zadate, a zatim ćemo ga upotrebiti kao potprogram da uokvirimo ekran. Evo najpre, potprograma koji je nešto pojednostavljen pretpostavkom da su dve tačke koje treba spajati nalaze na istoj horizontali ili vertikali. Ulazne veličine su brojevi X, Y, Z i T koji daju koordinate prve (X,Y) odnosno druge (Z,T) tačke; obzirom na uvedeno ograničenje mora da bude X=Z ili Y=T.

```
100 IF X=Z FOR I=Y TO T: DOT X,I:NEXT I: ELSE FOR I=X TO Z:
DOT I,Y:NEXT I
110 RETURN
```

U potprogramu, očigledno, ima i novina: napisana je jedna linija koja se sastoji od više naredbi razdvojenih znakom : Na ovaj način možemo da postupamo kada god želimo da štedimo memoriju ili kada nam se čini da grupu naredbi treba „spakovati" u isti red pošto predstavljaju logičnu celinu. Postoje ipak dva razloga zbog kojih čitav program ne može

da bude smešten u jednu programsku liniju maksimalna dužina linije koju računar „galaksija" može da primi je četiri reda a naredba koja se ne nalazi na početku neke linije ne može da se dosegne pomoću naredbi GOTO i CALL.

Evo najzad i programa koji će uokviriti ekran koristeći dati potprogram

```
 5 HOME
10 X=0  Y=0  Z=63  T=0  CALL 100
20 Y=47  T=47  CALL 100
30 X=63  Y=0  CALL 100
40 X=0  Y=0  Z=0  CALL 100
```

Vidi se da pre svakog poziva potprograma nisu postavljeni sadržaji svih promenljivih. Razlog je u tome što se ovi sadržaji u toku potprograma ne menjaju pa nema potrebe ponavljati iste naredbe i razbacivati memoriju.

Testiranje programa (koji je, jasno, povezan sa potprogramom) u prvom trenutku prolazi povoljno: računar uokviruje ekran. No, posle toga nastupa greška i to u naredbi RETURN. Zašto? Računar je četiri puta pozvao i izvršio potprogram i tako završio naredbu 40. Posle toga je jednostavno počeo sa izvršavanjem naredbe 100 koja sledi iza nje, još jednom povukao istu liniju i naišao na RETURN. Pošto potprogram nije pozivan, računar nije znao kuda da se vrati (RETURN = vrati se) pa je prijavio grešku.

Prva modifikacija u cilju rešavanja ovog problema je da se doda naredba 50 STOP (sećate li se naredbe STOP?). Solucija je dobra samo na prvi pogled: kada računar naiđe na STOP on prestaje sa izvršavanjem programa i izdaje poruku READY kvareći tako jedan deo teškom mukom nacrtanog okvira. Zato ćemo dodati naredbu 50 GOTO 50 koja navodi računar da se neprekidno vrti u „mrtvoj petlji" ne kvareći izgled ekrana (ovo je takozvano „dinamičko zaustavljanje"). Tako možemo da se divimo uokvirenom ekranu i pritisnemo taster **BRK** tek kada nam on dosadi.

U okviru potprograma može da se pozove sledeći potprogram, baš kao što u okviru jednog ciklusa (petlje) može da se započne drugi, ali je broj ovakvih poziva ograničen na 13. U praksi se sigurno neće događati da vam je potrebno više od trinaest nivoa potprograma, ali morate da pazite na početničke greške poput ove:

```
10 CALL 100

100 A=
150 GOTO 10
```

Na ovaj način računar neće nikada nailaziti na naredbu RETURN, pa će biti u situaciji da stalno poziva nove i nove potprograme sve dok ne dođe do greške. Ovakvu grešku je često teško otkriti: program dobro radi jedno vreme a onda u nepravilnim intervalima na ekranu ugledamo poruku HOW?

15. Ispravljanje programa — EDIT

Jedini način za ispravljanje grešaka u programu koji smo do sada upoznali bio je prekucavanje linije što je neekonomična i neobično nervirajuća solucija (u to ste se i sami uverili, pogotovu ako ste pisali neki program koji koristi naredbu PRINT AT). Računar „galaksija" poseduje i specijalnu komandu EDIT koja je namenjena ispravkama. Ona ne sme da se nađe u programu (blisko je pameti da programi mogu da se edituju samo u komandnom modu) i treba da je prati linijski broj naredbi koja se edituje. Pritiskom na **RET** računar briše ekran i u njegovim prvim redovima ispisuje liniju koju smo poželeli da menjamo. Kurzor se nalazi na njenom kraju. Koristeći tastere ← i →, možemo da ga pokrećemo u okviru naredbe sve dok ne dođemo do mesta na kome treba da ispravimo grešku. Ako želimo da izbacimo slovo koje se nalazi desno od kurzora, treba jednostavno da pritisnemo taster **DEL**. Ukoliko želimo da ubacimo novi deo programske linije, jednostavno ga otkucamo i on će da bude umetnut počevši od tekuće pozicije kurzora. Pri pomeranju kurzora i brisanju pojedinih delova programske linije, taster **REPT** (autorepeat) dolazi do punog izražaja.

Kada smo završili sa ispravkama neke linije (nema nikakve prepreke da u toku ispravki nekoliko puta menjamo isti deo linije), možemo da pritisnemo **RET** i tako se saglasimo sa izvršenim izmenama ili da pritisnemo **BRK** i tako naložimo računaru da ignoriše izmene i ostavi liniju u stanju u kom je bila pre nego što je editovanje započeto.

Pri editovanju se pojavljuje i jedan artefakt: nema prepreke da promenimo i sam broj programske linije pod uslovom da u okviru njega ne ostavljamo razmake (broj linije 10 02 nije dopušten). Kada pritisnemo **RET**, računar će u memoriju uneti novu programsku liniju dok stara neće biti ni promenjena ni izbrisana. Ovo pruža mogućnost umnožavanja jedne naredbe u okviru programa bez potrebe da se ona svaki put iznova kuca.

16. Numerički niz A (I)

Do sada smo upoznali promenljive A, B, C ... Y i Z pomoću kojih smeštamo podatke u memoriju računara. Ovi podaci su međusobno logički odvojeni: promenljiva V, na primer, može da čuva podatak o brzini nekog objekta, F o sili koja deluje na njega, T o vremenu i slično. Ponekad smo, međutim, suočeni sa potrebom da obradimo grupu međusobno srodnih podataka. Želimo, na primer, da sortiramo podatke po veličini. Besmisleno je da koristimo promenljive A, B ... obzirom da bismo u programu morali da navedemo veliku grupu naredbi koja bi poredila sadržaj svake promenljive sa sadržajima svih ostalih. Osim toga, šta da radimo ako treba da sortiramo više od 26 podataka?

Računar „galaksija" može da radi sa numeričkim nizom A(I) (ne treba ga mešati sa promenljivom A). Elementi ovog niza su A(0), A(1), A(2) itd. — njihov broj je ograničen jedino veličinom memorije. Iako ćemo mapu memorije objaviti tek docnije, potrebno je da već sada u grubim crtama objasnimo razmeštaj elemenata niza A(I) da bi se shvatila njegova promenljiva veličina. Zamislimo da se od početka memorije prema kraju smešta BASIC program, a od kraja memorije prema početku elementi niza A(I). Svaki element (A(0), A(1) ...) zauzima po četiri bajta. Jasno je da se negde na sredini memorije niz A(I) i BASIC program „susreću". Računar smatra da je BASIC program značajniji i ograničava niz A(I) tako da popunjava samo slobodan deo memorije. Ukoliko pokušamo da menjamo vrednost nekog elementa niza A(I) koji bi promenio bejzik program, računar će prekinuti sa radom i izdati poruku SORRY.

Kako da saznamo sa koliko elemenata niza A(I) raspolažemo? Jedan od načina je da otkucamo jednostavan program

```
10 I=0
20 A(I)=A(I)
30 I=I+1
40 GOTO 20
```

i startujemo ga sa RUN. Kada se na ekranu pojavi poruka SORRY, otkucamo PRINT I i saznamo broj prvog nepostojećeg elementa niza. Ovaj način nije naročito dobar, ponajviše zato što u postojeći bejzik program treba da dodajemo nove naredbe koje ga nepotrebno produžavaju i tako smanjuju broj članova niza. Zato je dizajnirana naredba nazvana MEM. PRINT MEM, na primer, prikazuje na ekranu broj bajtova memorije koji su slobodni za prijem niz A(I). Kako svaki element ovog niza zauzima po četiri bajta, PRINT MEM/4 će dati maksimalan broj elemenata koji nam stoji na raspolaganju.

Evo, na posletku, i programa koji će olakšati razumevanje rada sa nizom A(I). Program uređuje grupu brojeva u rastući niz i prikazuje dobijene rezultate. Korišćen je umereno dobar algoritam (postoje, naravno, i daleko bolji, ali su manje razumljivi za početnike) pa program unekoliko ilustruje brzinu rada računara „galaksija".

```
10 PRINT "KOLIKO IMA BROJEVA"; INPUT N
20 FOR I=1 TO N: INPUT A(I): NEXT I
30 FOR I=1 TO N-1: FOR J=I+1 TO N
40 IF A(I)<A(J) GOTO 60
50 P=A(I): A(I)=A(J): A(J)=P
60 NEXT J: NEXT I
70 FOR I=1 TO N: PRINT A(I),A(I): NEXT I
```